



# Arm<sup>®</sup> Cortex<sup>®</sup>-X925 Core

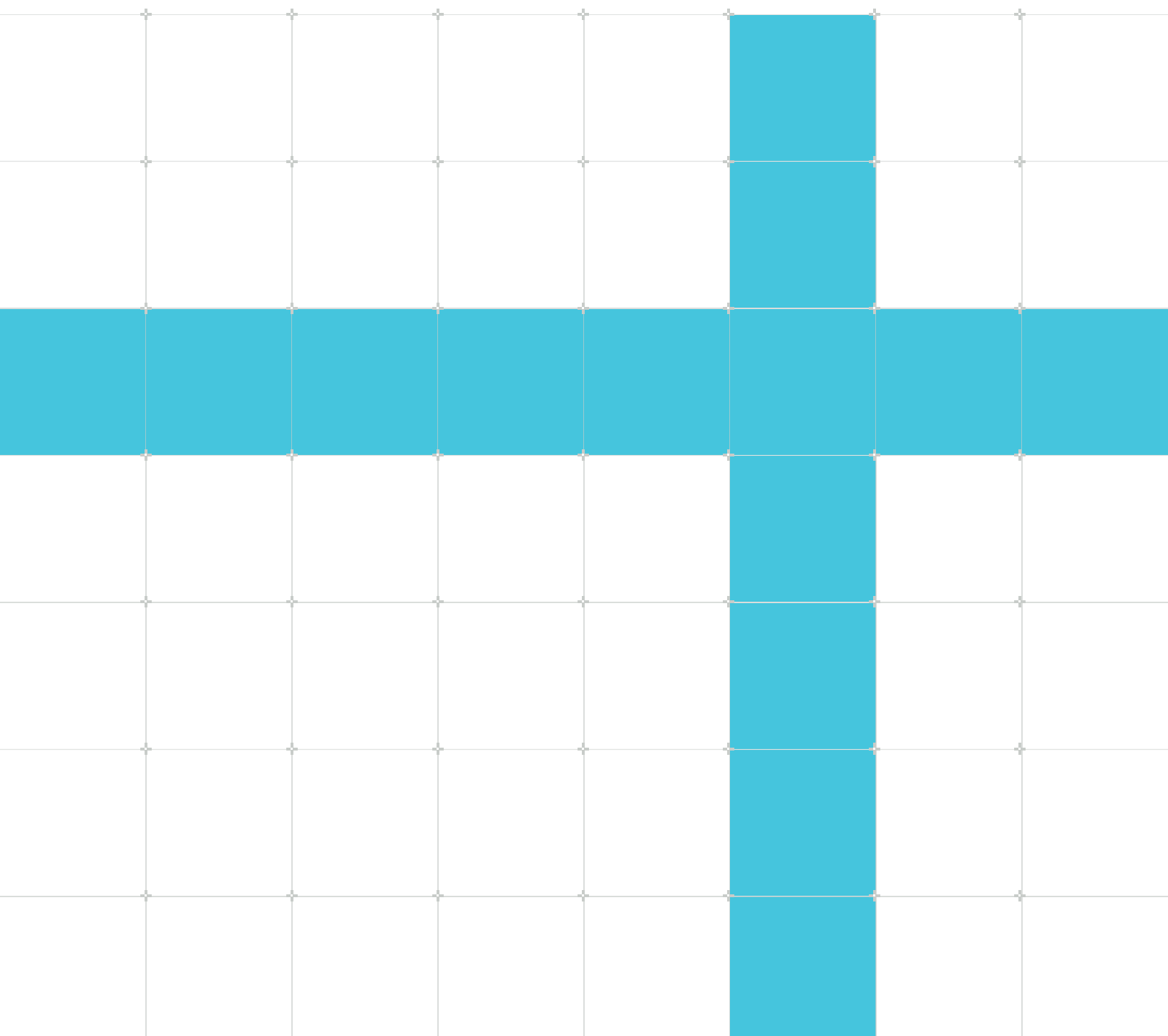
Revision r0p2

## Technical Reference Manual

**Non-Confidential**

**Issue 06**

Copyright © 2023–2025 Arm Limited (or its affiliates). 102807\_0002\_06\_en  
All rights reserved.



# Arm® Cortex®-X925 Core Technical Reference Manual

This document is Non-Confidential.

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted [Arm's Proprietary Notice](#) found at the end of this document.

This document (102807\_0002\_06\_en) was issued on 2025-05-22. There might be a later issue at <https://developer.arm.com/documentation/102807>

The product revision is r0p2.

See also: [Proprietary Notice](#) | [Product and document information](#) | [Useful resources](#)

## Start reading

If you prefer, you can skip to [the start of the content](#).

## Intended audience

This manual is for system designers, system integrators, and programmers who are designing or programming a *System on Chip* (SoC) that uses an Arm core.

## Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

# Contents

|                                                    |           |
|----------------------------------------------------|-----------|
| <b>1. The Cortex®-X925 core.....</b>               | <b>17</b> |
| 1.1 Cortex®-X925 core features.....                | 17        |
| 1.2 Cortex®-X925 core configuration options.....   | 19        |
| 1.3 DSU-120 dependent features.....                | 20        |
| 1.4 Supported standards and specifications.....    | 20        |
| 1.5 Test features.....                             | 27        |
| 1.6 Design tasks.....                              | 28        |
| 1.7 Product revisions.....                         | 29        |
| <b>2. Technical overview.....</b>                  | <b>30</b> |
| 2.1 Core components.....                           | 30        |
| 2.2 Interfaces.....                                | 34        |
| 2.3 Programmer's model.....                        | 35        |
| <b>3. Clocks and resets.....</b>                   | <b>36</b> |
| <b>4. Power management.....</b>                    | <b>37</b> |
| 4.1 Voltage and power domains.....                 | 37        |
| 4.2 Architectural clock gating modes.....          | 39        |
| 4.2.1 Wait for Interrupt and Wait for Event.....   | 39        |
| 4.2.2 Low-power state behavior considerations..... | 40        |
| 4.3 Power control.....                             | 41        |
| 4.4 Core power modes.....                          | 41        |
| 4.4.1 On mode.....                                 | 43        |
| 4.4.2 Off mode.....                                | 43        |
| 4.4.3 Emulated off mode.....                       | 44        |
| 4.4.4 Full retention mode.....                     | 44        |
| 4.4.5 Debug recovery mode.....                     | 45        |
| 4.4.6 Warm reset mode.....                         | 46        |
| 4.5 Performance and power management.....          | 46        |
| 4.5.1 Maximum Power Mitigation Mechanism.....      | 47        |
| 4.5.2 Performance Defined Power.....               | 47        |
| 4.5.3 Dispatch block.....                          | 48        |

|                                                                                        |           |
|----------------------------------------------------------------------------------------|-----------|
| 4.6 Cortex®-X925 core powerup and powerdown sequence.....                              | 48        |
| 4.6.1 Managing RAS fault and error interrupts during the core powerdown procedure..... | 49        |
| 4.7 Debug over powerdown.....                                                          | 49        |
| <b>5. Memory management.....</b>                                                       | <b>50</b> |
| 5.1 Memory Management Unit components.....                                             | 50        |
| 5.2 Translation Lookaside Buffer entry content.....                                    | 51        |
| 5.3 Translation Lookaside Buffer match process.....                                    | 52        |
| 5.4 Translation table walks.....                                                       | 53        |
| 5.5 Hardware management of the Access flag and dirty state.....                        | 54        |
| 5.6 Responses.....                                                                     | 54        |
| 5.7 Memory behavior and supported memory types.....                                    | 56        |
| 5.8 Page-based hardware attributes.....                                                | 57        |
| <b>6. L1 instruction memory system.....</b>                                            | <b>59</b> |
| 6.1 L1 instruction cache behavior.....                                                 | 59        |
| 6.2 L1 instruction cache Speculative memory accesses.....                              | 60        |
| 6.3 Program flow prediction.....                                                       | 60        |
| 6.4 Instruction Prefetch.....                                                          | 62        |
| <b>7. L1 data memory system.....</b>                                                   | <b>63</b> |
| 7.1 L1 data cache behavior.....                                                        | 63        |
| 7.2 Write streaming mode.....                                                          | 64        |
| 7.3 Instruction implementation in the L1 data memory system.....                       | 65        |
| 7.4 Internal exclusive monitor.....                                                    | 66        |
| 7.5 Data prefetching.....                                                              | 66        |
| <b>8. L2 memory system.....</b>                                                        | <b>68</b> |
| 8.1 L2 cache.....                                                                      | 68        |
| 8.2 Support for memory types.....                                                      | 69        |
| 8.3 Transaction capabilities.....                                                      | 69        |
| <b>9. Direct access to internal memory.....</b>                                        | <b>70</b> |
| 9.1 L1 cache encodings.....                                                            | 71        |
| 9.1.1 L1 instruction tag RAM returned data.....                                        | 72        |
| 9.1.2 L1 instruction data RAM returned data.....                                       | 73        |
| 9.1.3 L1 instruction TLB returned data.....                                            | 74        |
| 9.1.4 L1 data tag RAM returned data.....                                               | 75        |

|                                                          |            |
|----------------------------------------------------------|------------|
| 9.1.5 L1 data data RAM returned data.....                | 76         |
| 9.1.6 L1 data TLB returned data.....                     | 77         |
| 9.2 L2 cache encodings.....                              | 79         |
| 9.2.1 L2 tag RAM returned data.....                      | 81         |
| 9.2.2 L2 data RAM returned data.....                     | 82         |
| 9.2.3 L2 TLB RAM returned data.....                      | 83         |
| 9.2.4 L2 Victim RAM returned data.....                   | 86         |
| <b>10. RAS Extension support.....</b>                    | <b>88</b>  |
| 10.1 Cache protection behavior.....                      | 89         |
| 10.2 Error containment.....                              | 90         |
| 10.3 Fault detection and reporting.....                  | 90         |
| 10.4 Error detection and reporting.....                  | 90         |
| 10.4.1 Error reporting and performance monitoring.....   | 91         |
| 10.5 Error injection.....                                | 91         |
| 10.6 AArch64 RAS registers.....                          | 92         |
| 10.7 External RAS registers.....                         | 93         |
| <b>11. Utility bus.....</b>                              | <b>94</b>  |
| 11.1 Base addresses for system components.....           | 94         |
| <b>12. GIC CPU interface.....</b>                        | <b>96</b>  |
| 12.1 Disable the GIC CPU interface.....                  | 96         |
| 12.2 AArch64 GIC system registers.....                   | 97         |
| <b>13. Advanced SIMD and floating-point support.....</b> | <b>100</b> |
| <b>14. Scalable Vector Extensions support.....</b>       | <b>101</b> |
| <b>15. System control.....</b>                           | <b>102</b> |
| 15.1 AArch64 Generic System Control registers.....       | 102        |
| <b>16. Debug.....</b>                                    | <b>108</b> |
| 16.1 Supported debug methods.....                        | 109        |
| 16.2 Debug register interfaces.....                      | 110        |
| 16.2.1 Core interfaces.....                              | 110        |
| 16.2.2 Effects of resets on debug registers.....         | 111        |
| 16.2.3 Breakpoints and watchpoints.....                  | 112        |

|                                                                      |            |
|----------------------------------------------------------------------|------------|
| 16.3 Debug events.....                                               | 112        |
| 16.4 Debug memory map and debug signals.....                         | 112        |
| 16.5 ROM table.....                                                  | 113        |
| 16.6 CoreSight component identification.....                         | 113        |
| 16.7 CTI register identification values.....                         | 114        |
| 16.8 AArch64 Debug registers.....                                    | 114        |
| 16.9 External Debug registers.....                                   | 116        |
| 16.10 External CoreROM registers.....                                | 117        |
| <b>17. Performance Monitors Extension support.....</b>               | <b>119</b> |
| 17.1 Performance monitors events.....                                | 119        |
| 17.2 Performance monitors interrupts.....                            | 133        |
| 17.3 External register access permissions.....                       | 133        |
| 17.4 AArch64 Performance Monitors registers.....                     | 134        |
| 17.5 External PMU registers summary.....                             | 137        |
| <b>18. Embedded Trace Extension support.....</b>                     | <b>142</b> |
| 18.1 Trace unit resources.....                                       | 143        |
| 18.2 Trace unit generation options.....                              | 143        |
| 18.3 Reset the trace unit.....                                       | 144        |
| 18.4 Program and read the trace unit registers.....                  | 145        |
| 18.5 Trace unit register interfaces.....                             | 147        |
| 18.6 Interaction with the Performance Monitoring Unit and Debug..... | 147        |
| 18.7 Embedded Trace Extension events.....                            | 148        |
| 18.8 AArch64 Trace unit registers.....                               | 148        |
| 18.9 External ETE registers.....                                     | 152        |
| <b>19. Trace Buffer Extension support.....</b>                       | <b>154</b> |
| 19.1 Program and read the trace buffer registers.....                | 154        |
| 19.2 Trace buffer register interface.....                            | 154        |
| <b>20. Activity Monitors Extension support.....</b>                  | <b>155</b> |
| 20.1 Activity monitors access.....                                   | 155        |
| 20.2 Activity monitors counters.....                                 | 156        |
| 20.3 Activity monitors events.....                                   | 156        |
| 20.4 AArch64 Activity Monitors registers summary.....                | 157        |
| 20.5 External AMU registers.....                                     | 158        |

|                                                                                             |            |
|---------------------------------------------------------------------------------------------|------------|
| <b>21. Statistical Profiling Extension support.....</b>                                     | <b>161</b> |
| 21.1 Statistical Profiling Extension events packet.....                                     | 162        |
| 21.2 Statistical Profiling Extension data source packet.....                                | 163        |
| 21.3 AArch64 Statistical Profiling Extension registers.....                                 | 163        |
| <br><b>A. AArch64 System registers.....</b>                                                 | <b>165</b> |
| A.1 AArch64 registers summary.....                                                          | 165        |
| A.1.1 IMP_CPUPPMPDPCR_EL1, Performance and Power Management PDP Control Register...         | 165        |
| A.1.2 IMP_CPUPPMCR_EL3, Global Performance and Power Management Configuration Register..... | 167        |
| A.1.3 IMP_CPUMPMCR_EL3, Global MPMM Control Register.....                                   | 169        |
| A.1.4 IMP_CPUPPMCR4_EL3, CPU Power Performance Management Control Register.....             | 171        |
| A.1.5 IMP_CPUPPMCR5_EL3, CPU Power Performance Management Control Register.....             | 173        |
| A.1.6 IMP_CPUPPMCR6_EL3, CPU Power Performance Management Control Register.....             | 174        |
| A.2 AArch64 Generic System Control registers summary.....                                   | 176        |
| A.2.1 ACTLR_EL1, Auxiliary Control Register (EL1).....                                      | 181        |
| A.2.2 AFSR0_EL1, Auxiliary Fault Status Register 0 (EL1).....                               | 183        |
| A.2.3 AFSR1_EL1, Auxiliary Fault Status Register 1 (EL1).....                               | 186        |
| A.2.4 AMAIR_EL1, Auxiliary Memory Attribute Indirection Register (EL1).....                 | 188        |
| A.2.5 LORID_EL1, LORegionID (EL1).....                                                      | 191        |
| A.2.6 IMP_CPUACTLR_EL1, CPU Auxiliary Control Register (EL1).....                           | 193        |
| A.2.7 IMP_CPUACTLR2_EL1, CPU Auxiliary Control Register 2 (EL1).....                        | 195        |
| A.2.8 IMP_CPUACTLR3_EL1, CPU Auxiliary Control Register 3 (EL1).....                        | 196        |
| A.2.9 IMP_CPUACTLR4_EL1, CPU Auxiliary Control Register 4 (EL1).....                        | 198        |
| A.2.10 IMP_CPUECTLR_EL1, CPU Extended Control Register.....                                 | 200        |
| A.2.11 IMP_CPUECTLR2_EL1, CPU Extended Control Register 2.....                              | 210        |
| A.2.12 IMP_CPUL2DIRTYLNCT_EL1, CPU L2 Dirty Line Count Register.....                        | 215        |
| A.2.13 IMP_CPUPWRCTLR_EL1, CPU Power Control Register.....                                  | 216        |
| A.2.14 IMP_ATCR_EL1, CPU Auxiliary Translation Control Register (EL1).....                  | 219        |
| A.2.15 IMP_CPUACTLR5_EL1, CPU Auxiliary Control Register 5 (EL1).....                       | 222        |
| A.2.16 IMP_CPUACTLR6_EL1, CPU Auxiliary Control Register 6 (EL1).....                       | 223        |
| A.2.17 IMP_CPUACTLR7_EL1, CPU Auxiliary Control Register 7 (EL1).....                       | 225        |
| A.2.18 IMP_CPUACTLR10_EL1, CPU Auxiliary Control Register 10 (EL1).....                     | 227        |
| A.2.19 IMP_CPUACTLR8_EL1, CPU Auxiliary Control Register 8 (EL1).....                       | 229        |
| A.2.20 IMP_CPUACTLR9_EL1, CPU Auxiliary Control Register 9 (EL1).....                       | 230        |
| A.2.21 IMP_CPUACTLR11_EL1, CPU Auxiliary Control Register 11 (EL1).....                     | 232        |
| A.2.22 AIDR_EL1, Auxiliary ID Register.....                                                 | 234        |

|                                                                                            |     |
|--------------------------------------------------------------------------------------------|-----|
| A.2.23 FPCR, Floating-point Control Register.....                                          | 235 |
| A.2.24 ACTLR_EL2, Auxiliary Control Register (EL2).....                                    | 240 |
| A.2.25 HACR_EL2, Hypervisor Auxiliary Control Register.....                                | 243 |
| A.2.26 AFSR0_EL2, Auxiliary Fault Status Register 0 (EL2).....                             | 245 |
| A.2.27 AFSR1_EL2, Auxiliary Fault Status Register 1 (EL2).....                             | 247 |
| A.2.28 AMAIR_EL2, Auxiliary Memory Attribute Indirection Register (EL2).....               | 250 |
| A.2.29 IMP_ATCR_EL2, CPU Auxiliary Translation Control Register (EL2).....                 | 253 |
| A.2.30 IMP_AVTCR_EL2, CPU Virtualization Auxiliary Translation Control Register (EL2)..... | 255 |
| A.2.31 ACTLR_EL3, Auxiliary Control Register (EL3).....                                    | 257 |
| A.2.32 AFSR0_EL3, Auxiliary Fault Status Register 0 (EL3).....                             | 260 |
| A.2.33 AFSR1_EL3, Auxiliary Fault Status Register 1 (EL3).....                             | 262 |
| A.2.34 AMAIR_EL3, Auxiliary Memory Attribute Indirection Register (EL3).....               | 263 |
| A.2.35 RMR_EL3, Reset Management Register (EL3).....                                       | 265 |
| A.2.36 IMP_CPUL2SDIRTYLNCT_EL3, CPU L2 Secure Dirty Line Count Register.....               | 266 |
| A.2.37 IMP_CPUACTLR_EL3, CPU Auxiliary Control Register (EL3).....                         | 268 |
| A.2.38 IMP_ATCR_EL3, CPU Auxiliary Translation Control Register (EL3).....                 | 269 |
| A.2.39 IMP_CPUPSELR_EL3, Selected Instruction Private Select Register.....                 | 271 |
| A.2.40 IMP_CPUPCR_EL3, Selected Instruction Private Control Register.....                  | 273 |
| A.2.41 IMP_CPUPOR_EL3, Selected Instruction Private Opcode Register.....                   | 274 |
| A.2.42 IMP_CPUPMR_EL3, Selected Instruction Private Mask Register.....                     | 276 |
| A.2.43 IMP_CPUPOR2_EL3, Selected Instruction Private Opcode Register 2.....                | 277 |
| A.2.44 IMP_CPUPMR2_EL3, Selected Instruction Private Mask Register 2.....                  | 279 |
| A.2.45 IMP_CPUPFR_EL3, Selected Instruction Private Flag Register.....                     | 280 |
| A.3 AArch64 Debug registers summary.....                                                   | 282 |
| A.3.1 IMP_IDATA0_EL3, Instruction Register 0.....                                          | 284 |
| A.3.2 IMP_IDATA1_EL3, Instruction Register 1.....                                          | 285 |
| A.3.3 IMP_IDATA2_EL3, Instruction Register 2.....                                          | 286 |
| A.3.4 IMP_DDATA0_EL3, Data Register 0.....                                                 | 287 |
| A.3.5 IMP_DDATA1_EL3, Data Register 1.....                                                 | 289 |
| A.3.6 IMP_DDATA2_EL3, Data Register 2.....                                                 | 290 |
| A.4 AArch64 System instructions summary.....                                               | 291 |
| A.4.1 SYS_IMP_RAMINDEX, RAM Index.....                                                     | 292 |
| A.5 AArch64 Identification registers summary.....                                          | 293 |
| A.5.1 MIDR_EL1, Main ID Register.....                                                      | 294 |
| A.5.2 MPIDR_EL1, Multiprocessor Affinity Register.....                                     | 296 |
| A.5.3 REVIDR_EL1, Revision ID Register.....                                                | 298 |



|                                                                                            |     |
|--------------------------------------------------------------------------------------------|-----|
| A.5.4 ID_AA64PFR0_EL1, AArch64 Processor Feature Register 0.....                           | 300 |
| A.5.5 ID_AA64PFR1_EL1, AArch64 Processor Feature Register 1.....                           | 303 |
| A.5.6 ID_AA64ZFR0_EL1, SVE Feature ID Register 0.....                                      | 306 |
| A.5.7 ID_AA64DFR0_EL1, AArch64 Debug Feature Register 0.....                               | 309 |
| A.5.8 ID_AA64DFR1_EL1, AArch64 Debug Feature Register 1.....                               | 311 |
| A.5.9 ID_AA64AFR0_EL1, AArch64 Auxiliary Feature Register 0.....                           | 313 |
| A.5.10 ID_AA64AFR1_EL1, AArch64 Auxiliary Feature Register 1.....                          | 315 |
| A.5.11 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0.....                 | 316 |
| A.5.12 ID_AA64ISAR1_EL1, AArch64 Instruction Set Attribute Register 1.....                 | 320 |
| A.5.13 ID_AA64ISAR2_EL1, AArch64 Instruction Set Attribute Register 2.....                 | 323 |
| A.5.14 ID_AA64MMFR0_EL1, AArch64 Memory Model Feature Register 0.....                      | 326 |
| A.5.15 ID_AA64MMFR1_EL1, AArch64 Memory Model Feature Register 1.....                      | 329 |
| A.5.16 ID_AA64MMFR2_EL1, AArch64 Memory Model Feature Register 2.....                      | 332 |
| A.5.17 ID_AA64MMFR3_EL1, AArch64 Memory Model Feature Register 3.....                      | 335 |
| A.5.18 MPAMIDR_EL1, MPAM ID Register (EL1).....                                            | 338 |
| A.5.19 IMP_CPUCFR_EL1, CPU Configuration Register.....                                     | 340 |
| A.5.20 CLIDR_EL1, Cache Level ID Register.....                                             | 342 |
| A.5.21 GMID_EL1, Multiple tag transfer ID Register.....                                    | 346 |
| A.5.22 CTR_EL0, Cache Type Register.....                                                   | 348 |
| A.5.23 DCZID_EL0, Data Cache Zero ID Register.....                                         | 350 |
| A.6 AArch64 Special-purpose registers summary.....                                         | 352 |
| A.7 AArch64 Performance Monitors registers summary.....                                    | 353 |
| A.7.1 PMMIR_EL1, Performance Monitors Machine Identification Register.....                 | 356 |
| A.7.2 PMCR_EL0, Performance Monitors Control Register.....                                 | 359 |
| A.7.3 PMCEID0_EL0, Performance Monitors Common Event Identification Register 0.....        | 364 |
| A.7.4 PMCEID1_EL0, Performance Monitors Common Event Identification Register 1.....        | 371 |
| A.8 AArch64 GIC system registers summary.....                                              | 378 |
| A.8.1 ICC_APOR0_EL1, Interrupt Controller Active Priorities Group 0 Registers.....         | 380 |
| A.8.2 ICV_APOR0_EL1, Interrupt Controller Virtual Active Priorities Group 0 Registers..... | 390 |
| A.8.3 ICC_AP1R0_EL1, Interrupt Controller Active Priorities Group 1 Registers.....         | 400 |
| A.8.4 ICV_AP1R0_EL1, Interrupt Controller Virtual Active Priorities Group 1 Registers..... | 414 |
| A.8.5 ICC_CTLR_EL1, Interrupt Controller Control Register (EL1).....                       | 424 |
| A.8.6 ICV_CTLR_EL1, Interrupt Controller Virtual Control Register.....                     | 428 |
| A.8.7 ICH_APOR0_EL2, Interrupt Controller Hyp Active Priorities Group 0 Registers.....     | 432 |
| A.8.8 ICH_AP1R0_EL2, Interrupt Controller Hyp Active Priorities Group 1 Registers.....     | 439 |
| A.8.9 ICH_VTR_EL2, Interrupt Controller VGIC Type Register.....                            | 447 |

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| A.8.10 ICH_LR0_EL2, Interrupt Controller List Registers.....                   | 449 |
| A.8.11 ICH_LR1_EL2, Interrupt Controller List Registers.....                   | 453 |
| A.8.12 ICH_LR2_EL2, Interrupt Controller List Registers.....                   | 456 |
| A.8.13 ICH_LR3_EL2, Interrupt Controller List Registers.....                   | 460 |
| A.8.14 ICC_CTLR_EL3, Interrupt Controller Control Register (EL3).....          | 464 |
| A.9 AArch64 Generic Timer registers summary.....                               | 468 |
| A.10 AArch64 Other system control registers summary.....                       | 470 |
| A.11 AArch64 Activity Monitors registers summary.....                          | 471 |
| A.11.1 AMCFGR_ELO, Activity Monitors Configuration Register.....               | 472 |
| A.11.2 AMCGCR_ELO, Activity Monitors Counter Group Configuration Register..... | 475 |
| A.11.3 AMEVCNTR00_ELO, Activity Monitors Event Counter Registers 0.....        | 477 |
| A.11.4 AMEVCNTR01_ELO, Activity Monitors Event Counter Registers 0.....        | 479 |
| A.11.5 AMEVCNTR02_ELO, Activity Monitors Event Counter Registers 0.....        | 481 |
| A.11.6 AMEVCNTR03_ELO, Activity Monitors Event Counter Registers 0.....        | 483 |
| A.11.7 AMEVTYPER00_ELO, Activity Monitors Event Type Registers 0.....          | 485 |
| A.11.8 AMEVTYPER01_ELO, Activity Monitors Event Type Registers 0.....          | 488 |
| A.11.9 AMEVTYPER02_ELO, Activity Monitors Event Type Registers 0.....          | 490 |
| A.11.10 AMEVTYPER03_ELO, Activity Monitors Event Type Registers 0.....         | 492 |
| A.11.11 AMEVCNTR10_ELO, Activity Monitors Event Counter Registers 1.....       | 494 |
| A.11.12 AMEVCNTR11_ELO, Activity Monitors Event Counter Registers 1.....       | 496 |
| A.11.13 AMEVCNTR12_ELO, Activity Monitors Event Counter Registers 1.....       | 499 |
| A.11.14 AMEVTYPER10_ELO, Activity Monitors Event Type Registers 1.....         | 501 |
| A.11.15 AMEVTYPER11_ELO, Activity Monitors Event Type Registers 1.....         | 503 |
| A.11.16 AMEVTYPER12_ELO, Activity Monitors Event Type Registers 1.....         | 506 |
| A.12 AArch64 Trace unit registers summary.....                                 | 508 |
| A.12.1 TRCIDR8, ID Register 8.....                                             | 512 |
| A.12.2 TRCIMSPECO, IMP DEF Register 0.....                                     | 514 |
| A.12.3 TRCIDR10, ID Register 10.....                                           | 516 |
| A.12.4 TRCIDR11, ID Register 11.....                                           | 518 |
| A.12.5 TRCIDR12, ID Register 12.....                                           | 520 |
| A.12.6 TRCIDR13, ID Register 13.....                                           | 521 |
| A.12.7 TRCIDR0, ID Register 0.....                                             | 523 |
| A.12.8 TRCIDR1, ID Register 1.....                                             | 526 |
| A.12.9 TRCIDR2, ID Register 2.....                                             | 528 |
| A.12.10 TRCIDR3, ID Register 3.....                                            | 530 |
| A.12.11 TRCIDR4, ID Register 4.....                                            | 533 |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| A.12.12 TRCIDR5, ID Register 5.....                                            | 535        |
| A.12.13 TRCCIDCVR0, Context Identifier Comparator Value Registers <n>.....     | 537        |
| A.13 AArch64 Memory Partitioning and Monitoring registers summary.....         | 540        |
| A.13.1 MPAMVPMV_EL2, MPAM Virtual Partition Mapping Valid Register.....        | 540        |
| A.13.2 MPAMVPM0_EL2, MPAM Virtual PARTID Mapping Register 0.....               | 542        |
| A.13.3 MPAMVPM1_EL2, MPAM Virtual PARTID Mapping Register 1.....               | 545        |
| A.14 AArch64 RAS registers summary.....                                        | 547        |
| A.14.1 ERRIDR_EL1, Error Record ID Register.....                               | 548        |
| A.14.2 ERRSELR_EL1, Error Record Select Register.....                          | 549        |
| A.14.3 ERXFR_EL1, Selected Error Record Feature Register.....                  | 552        |
| A.14.4 ERXCTLR_EL1, Selected Error Record Control Register.....                | 555        |
| A.14.5 ERXSTATUS_EL1, Selected Error Record Primary Status Register.....       | 559        |
| A.14.6 ERXADDR_EL1, Selected Error Record Address Register.....                | 565        |
| A.14.7 ERXPFGF_EL1, Selected Pseudo-fault Generation Feature Register.....     | 568        |
| A.14.8 ERXPFGCTL_EL1, Selected Pseudo-fault Generation Control Register.....   | 572        |
| A.14.9 ERXPFGCDN_EL1, Selected Pseudo-fault Generation Countdown Register..... | 577        |
| A.14.10 ERXMISCO_EL1, Selected Error Record Miscellaneous Register 0.....      | 580        |
| A.14.11 ERXMISC1_EL1, Selected Error Record Miscellaneous Register 1.....      | 586        |
| A.14.12 ERXMISC2_EL1, Selected Error Record Miscellaneous Register 2.....      | 589        |
| A.14.13 ERXMISC3_EL1, Selected Error Record Miscellaneous Register 3.....      | 591        |
| A.15 AArch64 Statistical Profiling Extension registers summary.....            | 594        |
| A.15.1 PMSNEVFR_EL1, Sampling Inverted Event Filter Register.....              | 595        |
| A.15.2 PMSEVFR_EL1, Sampling Event Filter Register.....                        | 605        |
| A.15.3 PMSIDR_EL1, Sampling Profiling ID Register.....                         | 614        |
| A.15.4 PMBIDR_EL1, Profiling Buffer ID Register.....                           | 617        |
| A.16 AArch64 Trace Buffer Extension registers summary.....                     | 619        |
| <b>B. External registers.....</b>                                              | <b>620</b> |
| B.1 External CoreROM registers summary.....                                    | 620        |
| B.1.1 COREROM_ROMENTRY0, Core ROM table Entry 0.....                           | 621        |
| B.1.2 COREROM_ROMENTRY1, Core ROM table Entry 1.....                           | 622        |
| B.1.3 COREROM_ROMENTRY2, Core ROM table Entry 2.....                           | 623        |
| B.1.4 COREROM_ROMENTRY3, Core ROM table Entry 3.....                           | 625        |
| B.1.5 COREROM_AUTHSTATUS, Core ROM table Authentication Status Register.....   | 626        |
| B.1.6 COREROM_DEVARCH, Core ROM table Device Architecture Register.....        | 628        |
| B.1.7 COREROM_DEVTYPE, Core ROM table Device Type Register.....                | 629        |

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| B.1.8 COREROM_PIDR4, Core ROM table Peripheral Identification Register 4.....       | 630 |
| B.1.9 COREROM_PIDR0, Core ROM table Peripheral Identification Register 0.....       | 632 |
| B.1.10 COREROM_PIDR1, Core ROM table Peripheral Identification Register 1.....      | 633 |
| B.1.11 COREROM_PIDR2, Core ROM table Peripheral Identification Register 2.....      | 634 |
| B.1.12 COREROM_PIDR3, Core ROM table Peripheral Identification Register 3.....      | 635 |
| B.1.13 COREROM_CIDR0, Core ROM table Component Identification Register 0.....       | 637 |
| B.1.14 COREROM_CIDR1, Core ROM table Component Identification Register 1.....       | 638 |
| B.1.15 COREROM_CIDR2, Core ROM table Component Identification Register 2.....       | 639 |
| B.1.16 COREROM_CIDR3, Core ROM table Component Identification Register 3.....       | 640 |
| B.2 External PPM registers summary.....                                             | 642 |
| B.2.1 CPUPPMCR, Global Performance and Power Management Configuration Register..... | 642 |
| B.2.2 CPUMPMCR, Global MPMM Control Register.....                                   | 644 |
| B.2.3 CPUPPMPDPCR, Performance and Power Management PDP Control Register.....       | 645 |
| B.2.4 CPUPPMCR4, Power Performance Management Register.....                         | 647 |
| B.2.5 CPUPPMCR5, Power Performance Management Register.....                         | 648 |
| B.2.6 CPUPPMCR6, Power Performance Management Register.....                         | 649 |
| B.3 External PMU registers summary.....                                             | 650 |
| B.3.1 PMPCSSR, Snapshot Program Counter Sample Register.....                        | 654 |
| B.3.2 PMCIDSSR, Snapshot CONTEXTIDR_EL1 Sample Register.....                        | 656 |
| B.3.3 PMCID2SSR, Snapshot CONTEXTIDR_EL2 Sample Register.....                       | 657 |
| B.3.4 PMSSSR, PMU Snapshot Status Register.....                                     | 658 |
| B.3.5 PMCCNTSR, PMU Cycle Counter Snapshot Register.....                            | 660 |
| B.3.6 PMCFGR, Performance Monitors Configuration Register.....                      | 661 |
| B.3.7 PMCR_ELO, Performance Monitors Control Register.....                          | 664 |
| B.3.8 PMCEID0, Performance Monitors Common Event Identification register 0.....     | 666 |
| B.3.9 PMCEID1, Performance Monitors Common Event Identification register 1.....     | 670 |
| B.3.10 PMCEID2, Performance Monitors Common Event Identification register 2.....    | 674 |
| B.3.11 PMCEID3, Performance Monitors Common Event Identification register 3.....    | 678 |
| B.3.12 PMSSCR, PMU Snapshot Capture Register.....                                   | 682 |
| B.3.13 PMMIR, Performance Monitors Machine Identification Register.....             | 683 |
| B.3.14 PMDEVARCH, Performance Monitors Device Architecture register.....            | 685 |
| B.3.15 PMDEVID, Performance Monitors Device ID register.....                        | 686 |
| B.3.16 PMDEVTYPE, Performance Monitors Device Type register.....                    | 688 |
| B.3.17 PMPIDR4, Performance Monitors Peripheral Identification Register 4.....      | 689 |
| B.3.18 PMPIDR0, Performance Monitors Peripheral Identification Register 0.....      | 691 |
| B.3.19 PMPIDR1, Performance Monitors Peripheral Identification Register 1.....      | 692 |

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| B.3.20 PMPIDR2, Performance Monitors Peripheral Identification Register 2..... | 694 |
| B.3.21 PMPIDR3, Performance Monitors Peripheral Identification Register 3..... | 695 |
| B.3.22 PMCIDR0, Performance Monitors Component Identification Register 0.....  | 697 |
| B.3.23 PMCIDR1, Performance Monitors Component Identification Register 1.....  | 698 |
| B.3.24 PMCIDR2, Performance Monitors Component Identification Register 2.....  | 699 |
| B.3.25 PMCIDR3, Performance Monitors Component Identification Register 3.....  | 701 |
| B.4 External Debug registers summary.....                                      | 702 |
| B.4.1 EDRCR, External Debug Reserve Control Register.....                      | 704 |
| B.4.2 EDPRCR, External Debug Power/Reset Control Register.....                 | 706 |
| B.4.3 MIDR_EL1, Main ID Register.....                                          | 708 |
| B.4.4 EDPFR, External Debug Processor Feature Register.....                    | 709 |
| B.4.5 EDDFR, External Debug Feature Register.....                              | 712 |
| B.4.6 EDDFR1, External Debug Feature Register 1.....                           | 714 |
| B.4.7 EDDEVARCH, External Debug Device Architecture Register.....              | 716 |
| B.4.8 EDDEVID2, External Debug Device ID register 2.....                       | 717 |
| B.4.9 EDDEVID1, External Debug Device ID Register 1.....                       | 719 |
| B.4.10 EDDEVID, External Debug Device ID register 0.....                       | 720 |
| B.4.11 EDDEVTYPE, External Debug Device Type register.....                     | 721 |
| B.4.12 EDPIDR4, External Debug Peripheral Identification Register 4.....       | 723 |
| B.4.13 EDPIDR0, External Debug Peripheral Identification Register 0.....       | 724 |
| B.4.14 EDPIDR1, External Debug Peripheral Identification Register 1.....       | 725 |
| B.4.15 EDPIDR2, External Debug Peripheral Identification Register 2.....       | 727 |
| B.4.16 EDPIDR3, External Debug Peripheral Identification Register 3.....       | 728 |
| B.4.17 EDCIDR0, External Debug Component Identification Register 0.....        | 730 |
| B.4.18 EDCIDR1, External Debug Component Identification Register 1.....        | 731 |
| B.4.19 EDCIDR2, External Debug Component Identification Register 2.....        | 732 |
| B.4.20 EDCIDR3, External Debug Component Identification Register 3.....        | 734 |
| B.5 External CTI registers summary.....                                        | 735 |
| B.5.1 CTIPIDR0, CTI Peripheral Identification Register 0.....                  | 735 |
| B.5.2 CTIPIDR1, CTI Peripheral Identification Register 1.....                  | 737 |
| B.6 External AMU registers summary.....                                        | 738 |
| B.6.1 AMEVCNTR00, Activity Monitors Event Counter Registers 0.....             | 739 |
| B.6.2 AMEVCNTR01, Activity Monitors Event Counter Registers 0.....             | 741 |
| B.6.3 AMEVCNTR02, Activity Monitors Event Counter Registers 0.....             | 743 |
| B.6.4 AMEVCNTR03, Activity Monitors Event Counter Registers 0.....             | 745 |
| B.6.5 AMEVCNTR10, Activity Monitors Event Counter Registers 1.....             | 746 |

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| B.6.6 AMEVCNTR11, Activity Monitors Event Counter Registers 1.....           | 748 |
| B.6.7 AMEVCNTR12, Activity Monitors Event Counter Registers 1.....           | 750 |
| B.6.8 AMEVTYPER00, Activity Monitors Event Type Registers 0.....             | 752 |
| B.6.9 AMEVTYPER01, Activity Monitors Event Type Registers 0.....             | 753 |
| B.6.10 AMEVTYPER02, Activity Monitors Event Type Registers 0.....            | 755 |
| B.6.11 AMEVTYPER03, Activity Monitors Event Type Registers 0.....            | 757 |
| B.6.12 AMEVTYPER10, Activity Monitors Event Type Registers 1.....            | 759 |
| B.6.13 AMEVTYPER11, Activity Monitors Event Type Registers 1.....            | 761 |
| B.6.14 AMEVTYPER12, Activity Monitors Event Type Registers 1.....            | 762 |
| B.6.15 AMCGCR, Activity Monitors Counter Group Configuration Register.....   | 764 |
| B.6.16 AMCFGR, Activity Monitors Configuration Register.....                 | 766 |
| B.6.17 AMIIDR, Activity Monitors Implementation Identification Register..... | 767 |
| B.6.18 AMDEVARCH, Activity Monitors Device Architecture Register.....        | 769 |
| B.6.19 AMDEVTYPE, Activity Monitors Device Type Register.....                | 770 |
| B.6.20 AMPIDR4, Activity Monitors Peripheral Identification Register 4.....  | 771 |
| B.6.21 AMPIDR0, Activity Monitors Peripheral Identification Register 0.....  | 773 |
| B.6.22 AMPIDR1, Activity Monitors Peripheral Identification Register 1.....  | 774 |
| B.6.23 AMPIDR2, Activity Monitors Peripheral Identification Register 2.....  | 775 |
| B.6.24 AMPIDR3, Activity Monitors Peripheral Identification Register 3.....  | 777 |
| B.6.25 AMCIDR0, Activity Monitors Component Identification Register 0.....   | 778 |
| B.6.26 AMCIDR1, Activity Monitors Component Identification Register 1.....   | 779 |
| B.6.27 AMCIDR2, Activity Monitors Component Identification Register 2.....   | 781 |
| B.6.28 AMCIDR3, Activity Monitors Component Identification Register 3.....   | 782 |
| B.7 External ETE registers summary.....                                      | 783 |
| B.7.1 TRCAUXCTLR, Auxiliary Control Register.....                            | 784 |
| B.7.2 TRCIDR8, ID Register 8.....                                            | 786 |
| B.7.3 TRCIDR9, ID Register 9.....                                            | 787 |
| B.7.4 TRCIDR10, ID Register 10.....                                          | 788 |
| B.7.5 TRCIDR11, ID Register 11.....                                          | 789 |
| B.7.6 TRCIDR12, ID Register 12.....                                          | 790 |
| B.7.7 TRCIDR13, ID Register 13.....                                          | 791 |
| B.7.8 TRCIMSPECO, IMP DEF Register 0.....                                    | 792 |
| B.7.9 TRCIDR0, ID Register 0.....                                            | 794 |
| B.7.10 TRCIDR1, ID Register 1.....                                           | 796 |
| B.7.11 TRCIDR2, ID Register 2.....                                           | 798 |
| B.7.12 TRCIDR3, ID Register 3.....                                           | 799 |

|                                                                                         |     |
|-----------------------------------------------------------------------------------------|-----|
| B.7.13 TRCIDR4, ID Register 4.....                                                      | 802 |
| B.7.14 TRCIDR5, ID Register 5.....                                                      | 804 |
| B.7.15 TRCIDR6, ID Register 6.....                                                      | 805 |
| B.7.16 TRCIDR7, ID Register 7.....                                                      | 807 |
| B.7.17 TRCSSCSR<n>, Single-shot Comparator Control Status Register <n> , n = 0 - 7..... | 808 |
| B.7.18 TRCITCTRL, Integration Mode Control Register.....                                | 811 |
| B.7.19 TRCCLAIMSET, Claim Tag Set Register.....                                         | 812 |
| B.7.20 TRCCLAIMCLR, Claim Tag Clear Register.....                                       | 814 |
| B.7.21 TRCDEVARCH, Device Architecture Register.....                                    | 815 |
| B.7.22 TRCDEVID2, Device Configuration Register 2.....                                  | 817 |
| B.7.23 TRCDEVID1, Device Configuration Register 1.....                                  | 818 |
| B.7.24 TRCDEVID, Device Configuration Register.....                                     | 819 |
| B.7.25 TRCDEVTYPE, Device Type Register.....                                            | 820 |
| B.7.26 TRCPIDR4, Peripheral Identification Register 4.....                              | 822 |
| B.7.27 TRCPIDR5, Peripheral Identification Register 5.....                              | 823 |
| B.7.28 TRCPIDR6, Peripheral Identification Register 6.....                              | 825 |
| B.7.29 TRCPIDR7, Peripheral Identification Register 7.....                              | 826 |
| B.7.30 TRCPIDR0, Peripheral Identification Register 0.....                              | 827 |
| B.7.31 TRCPIDR1, Peripheral Identification Register 1.....                              | 828 |
| B.7.32 TRCPIDR2, Peripheral Identification Register 2.....                              | 830 |
| B.7.33 TRCPIDR3, Peripheral Identification Register 3.....                              | 832 |
| B.7.34 TRCCIDR0, Component Identification Register 0.....                               | 833 |
| B.7.35 TRCCIDR1, Component Identification Register 1.....                               | 835 |
| B.7.36 TRCCIDR2, Component Identification Register 2.....                               | 836 |
| B.7.37 TRCCIDR3, Component Identification Register 3.....                               | 837 |
| B.8 External RAS registers summary.....                                                 | 839 |
| B.8.1 ERROFR, Error Record <n> Feature Register.....                                    | 839 |
| B.8.2 ERROCTLR, Error Record <n> Control Register.....                                  | 843 |
| B.8.3 ERROSTATUS, Error Record <n> Primary Status Register.....                         | 845 |
| B.8.4 ERROADDR, Error Record <n> Address Register.....                                  | 853 |
| B.8.5 ERR0MISCO, Error Record <n> Miscellaneous Register 0.....                         | 854 |
| B.8.6 ERR0MISC1, Error Record <n> Miscellaneous Register 1.....                         | 864 |
| B.8.7 ERR0MISC2, Error Record <n> Miscellaneous Register 2.....                         | 867 |
| B.8.8 ERR0MISC3, Error Record <n> Miscellaneous Register 3.....                         | 869 |
| B.8.9 ERROPFGF, Error Record <n> Pseudo-fault Generation Feature Register.....          | 872 |
| B.8.10 ERROPFGCTL, Error Record <n> Pseudo-fault Generation Control Register.....       | 875 |

B.8.11 ERROPFGCDN, Error Record <n> Pseudo-fault Generation Countdown Register.....878

**Proprietary Notice.....880**

**Product and document information..... 882**

Product status.....882

Revision history..... 882

Conventions.....885

**Useful resources..... 888**



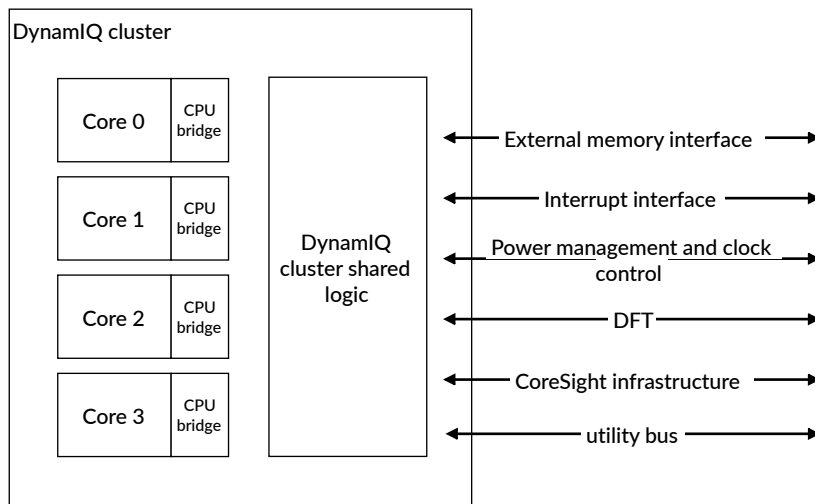
# 1. The Cortex®-X925 core

The Cortex®-X925 core is a high-performance and low-power product that implements the Arm®v9.2-A architecture. The Arm®v9.2-A architecture extends the architecture defined in the Arm®v8-A architectures up to Arm®v8.7-A.

The Cortex®-X925 core is implemented inside a DSU-120 DynamIQ™ cluster. It is connected to the *DynamIQ™ Shared Unit-120* that behaves as a full interconnect with L3 cache and snoop control. This connection configuration is also used in systems with different types of cores where the Cortex®-X925 core is the high-performance core.

The following figure shows an example configuration with four Cortex®-X925 cores in a DynamIQ™ cluster.

**Figure 1-1: Cortex®-X925 cores example configuration**



**Note**

- This manual applies to the Cortex®-X925 core only. Read this manual together with the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#) for detailed information about the DSU-120.
- This manual does not provide a complete list of registers. Read this manual together with the [Arm® Architecture Reference Manual for A-profile architecture](#).

## 1.1 Cortex®-X925 core features

You can use the Cortex®-X925 core in a standalone DynamIQ™ configuration where your homogeneous DSU-120 DynamIQ™ cluster includes one or more Cortex®-X925 cores. You can also use the Cortex®-X925 core as the high-performance core in a heterogeneous cluster.

Regardless of the cluster configuration, the Cortex®-X925 core always has the same features as described in the following lists.

## Core features

- Implementation of the Arm®v9.2-A A64 instruction set
- AArch64 Execution state at all Exception levels, EL0 to EL3
- *Memory Management Unit* (MMU)
- 40-bit *Physical Address* (PA) and 48-bit *Virtual Address* (VA)
- *Generic Interrupt Controller* (GIC) CPU interface to connect to an external interrupt Distributor
- Generic Timers interface that supports 64-bit count input from an external system counter
- Implementation of the *Reliability, Availability, and Serviceability* (RAS) Extension
- Implementation of the *Scalable Vector Extension* (SVE) with a 128-bit vector length and *Scalable Vector Extension 2* (SVE2)
- Integrated execution unit with *Advanced Single Instruction Multiple Data* (SIMD) and floating-point support
- *Activity Monitoring Unit* (AMU)
- Support for the optional Cryptographic Extension



The Cryptographic Extension is licensed separately.

---

## Cache features

- Separate L1 data and instruction caches
- Private, unified data and instruction L2 cache
- Error protection on L1 instruction and data caches, L2 cache, and *MMU Translation Cache* (MMU TC) with parity or *Error Correcting Code* (ECC) allowing *Single Error Correction and Double Error Detection* (SECCDED).
- Support for *Memory System Resource Partitioning and Monitoring* (MPAM)

## Debug features

- Arm®v9.2-A debug logic
- *Performance Monitoring Unit* (PMU)
- *Embedded Trace Extension* (ETE)
- *TRace Buffer Extension* (TRBE)
- *Statistical Profiling Extension* (SPE)
- Optional *Embedded Logic Analyzer* (ELA), ELA-600

**Note**

The ELA-600 is licensed separately.

---

## Related information

2. [Technical overview](#) on page 30

## 1.2 Cortex®-X925 core configuration options

You can choose the options that fit your implementation needs at build-time configuration.

The Cortex®-X925 core implementation options include:

### Cryptographic extension

You can configure your implementation with or without the Cryptographic Extension. The selected option applies to all cores in the cluster.

### L2 Data RAM ECC granule

You can configure the L2 Data RAM ECC granule to be 128 bits or 256 bits

### L2 cache size

You can configure the L2 cache to be 2048KB or 3072KB. The cores in the cluster can have different cache sizes.

### PMU event counters

You can configure the number of PMU events counters to be 6 or 31.

### CoreSight ELA

You can include support for integrating ELA-600 as a separate licensable product. This option can be configured on a per-core basis.

### Size of the ATB FIFO depth in the core ELA

You can configure the size of the AMBA® *Trace Bus* (ATB) FIFO to be 4, 8, 16, 32, or 64. This option can be configured on a per-core basis.

### Timing closure

You can configure the L2 data cache RAMs timing behavior. For more information, see *Cortex®-X925 configuration parameters* in the *Arm® Cortex®-X925 Core Configuration and Integration Manual*.

**Note**

The Cortex®-X925 cores in the DSU-120 cluster must have identical configurations, except for the L2 cache size.

---

For detailed configuration options and guidelines, see *RTL configuration process* in the *Arm® Cortex®-X925 Core Configuration and Integration Manual*.

## 1.3 DSU-120 dependent features

Some *DynamIQ™ Shared Unit-120* (DSU-120) features and behaviors depend on whether your licensed core supports a particular feature.

The following table describes which DSU-120 dependent features are supported in your Cortex®-X925 core.

**Table 1-1: Cortex®-X925 core features that have a dependency on the DSU-120**

| Feature                                            | Supported in the Cortex®-X925 core | Dependency on the DSU-120                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Direct connect                                     | No                                 | -                                                                                                                                                                                                                                                                                                          |
| Core included in a complex                         | No                                 | Affects the DSU-120 DynamIQ™ cluster configuration and external signals.                                                                                                                                                                                                                                   |
| Cryptographic Extension                            | Yes, as an option                  | Affects the external signals of the DSU-120.<br><br>For more information on MPMM, PDP, and the dispatch block signal, see <a href="#">4.5 Performance and power management</a> on page 46.                                                                                                                 |
| Maximum Power Mitigation Mechanism (MPMM)          | Yes                                |                                                                                                                                                                                                                                                                                                            |
| Performance Defined Power (PDP) feature            | Yes                                |                                                                                                                                                                                                                                                                                                            |
| DISPBLK <sub>y</sub>                               | Yes                                |                                                                                                                                                                                                                                                                                                            |
| Dispatch block signal                              |                                    |                                                                                                                                                                                                                                                                                                            |
| Statistical Profiling Extension (SPE) architecture | Yes                                |                                                                                                                                                                                                                                                                                                            |
| Physical Address (PA) width                        | 40-bit                             | Affects the CHI master and AXI master port bus widths.<br><br>For more details, see the following chapters of the <i>Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual</i> : <ul style="list-style-type: none"> <li>• <i>CHI master interface</i></li> <li>• <i>AXI master interface</i></li> </ul> |



- The Cryptographic Extension is supplied under a separate license.

## 1.4 Supported standards and specifications

The Cortex®-X925 core complies with the Arm®v9.2-A architecture. The Arm®v9.2-A architecture extends the architecture defined in the Arm®v8-A architectures up to Arm®v8.7-A. The

Cortex®-X925 core also complies with specific Arm®v8-A architecture extensions and supports interconnect, interrupt, timer, debug, and trace architectures.

The Cortex®-X925 core supports AArch64 at Exception levels EL0 to EL3.

Not all architectural features are implemented in the Cortex®-X925 core. The following tables show the implementation status of Arm®v8-A and Arm®v9-A features supported by the Cortex®-X925 core. There is a separate table for each version of the Arm®v8-A and Arm®v9-A architectures.



**Note**

- Not all Arm®v8-A and Arm®v9-A architectural features are listed in the following tables. For more information on all the architectural features, see the [Arm® Architecture Reference Manual for A-profile architecture](#).
- The Cortex®-X925 core is compatible with the architecture for the *DynamIQ™ Shared Unit-120*. See the *Supported standards and specifications* section in the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#) for a list of specific architectural versions and features supported by the DSU-120.

**Table 1-2: Implementation status of the Arm®v8.0-A features in the Cortex®-X925 core**

| Feature                 | Implemented                                               | Description                                                                                                                                                                                                                                                                                   |
|-------------------------|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEAT_PCSRv8             | No                                                        | PC Sample-based Profiling Extension                                                                                                                                                                                                                                                           |
| Cryptographic Extension | Yes<br>Configurable                                       | For more information and additional cryptographic register descriptions, see the <a href="#">Arm® Cortex®-X925 Core Cryptographic Extension Technical Reference Manual</a> .<br><br>This extension is licensed separately and access to the documentation is restricted by contract with Arm. |
| FEAT_SHA1               | Yes                                                       | Advanced SIMD SHA1 instructions                                                                                                                                                                                                                                                               |
| FEAT_SHA256             | Configurable                                              | Advanced SIMD SHA256 instructions                                                                                                                                                                                                                                                             |
| FEAT_AES                |                                                           | Advanced SIMD AES instructions                                                                                                                                                                                                                                                                |
| FEAT_PMULL              | Supported as part of the Arm®v8-A Cryptographic Extension | Advanced SIMD PMULL instructions                                                                                                                                                                                                                                                              |
| FEAT_DoubleLock         | No                                                        | Double Lock                                                                                                                                                                                                                                                                                   |
| FEAT_CP15SDISABLE2      | No                                                        | CP15DISABLE2                                                                                                                                                                                                                                                                                  |
| FEAT_FP                 | Yes                                                       | Floating point extension                                                                                                                                                                                                                                                                      |
| FEAT_AdvSIMD            | Yes                                                       | Advanced SIMD Extension<br><br>For more information and register descriptions, see <a href="#">13. Advanced SIMD and floating-point support</a> on page 100.                                                                                                                                  |
| FEAT_CRC32              | Yes                                                       | CRC32 instructions                                                                                                                                                                                                                                                                            |
| FEAT_PMUV3              | Yes                                                       | PMU extension version 3                                                                                                                                                                                                                                                                       |
| FEAT_nTLBPA             | Yes                                                       | No intermediate caching by output address in TLB                                                                                                                                                                                                                                              |
| FEAT_SB                 | Yes                                                       | Speculation barrier                                                                                                                                                                                                                                                                           |
| FEAT_SSBS               | Yes                                                       | Speculative Store Bypass Safe Instruction                                                                                                                                                                                                                                                     |

| Feature       | Implemented | Description                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEAT_SSBS2    | Yes         | MRS and MSR instructions for SSBS version 2                                                                                                                                                                                                                                                                                                                                                                   |
| FEAT_CSV2     | Yes         | Cache Speculation Variant 2                                                                                                                                                                                                                                                                                                                                                                                   |
| FEAT_CSV2_1p1 | No          | Cache Speculation Variant 2 version 1.1                                                                                                                                                                                                                                                                                                                                                                       |
| FEAT_CSV2_1p2 | No          | Cache Speculation Variant 2 version 1.2                                                                                                                                                                                                                                                                                                                                                                       |
| FEAT_CSV2_2   | Yes         | Cache Speculation Variant 2 version 2                                                                                                                                                                                                                                                                                                                                                                         |
| FEAT_CSV3     | Yes         | Cache Speculation Variant 3                                                                                                                                                                                                                                                                                                                                                                                   |
| FEAT_SPECRES  | Yes         | Speculation restriction instructions                                                                                                                                                                                                                                                                                                                                                                          |
| FEAT_DGH      | Yes         | Data Gathering Hint                                                                                                                                                                                                                                                                                                                                                                                           |
| FEAT_ETS      | Yes         | Enhanced Translation Synchronization                                                                                                                                                                                                                                                                                                                                                                          |
| FEAT_ECBHB    | Yes         | <p><i>Exploitative Control using Branch History Buffer</i> information between exception levels</p> <p>The branch history information created in a context before an exception to a higher exception level, using AArch64, cannot be used by code before that exception. This prevents exploitative control of the execution of any indirect branches in code in a different context after the exception.</p> |

**Table 1-3: Implementation status of the Arm®v8.1-A features in the Cortex®-X925 core**

| Feature        | Implemented | Description                                                           |
|----------------|-------------|-----------------------------------------------------------------------|
| FEAT_LSE       | Yes         | Large System Extensions                                               |
| FEAT_RDM       | Yes         | Rounding double multiply accumulate                                   |
| FEAT_HPDS      | Yes         | Hierarchical permission disables in translation tables                |
| FEAT_VHE       | Yes         | Virtualization Host Extensions                                        |
| FEAT_PAN       | Yes         | Privileged access-never                                               |
| FEAT_LOR       | Yes         | Limited ordering regions                                              |
| FEAT_HAFDBS    | Yes         | Hardware updates to access flag and dirty state in translation tables |
| FEAT_VMID16    | Yes         | 16-bit VMID                                                           |
| FEAT_PMUv3p1   | Yes         | PMU extensions version 3.1                                            |
| FEAT_Debugv8p1 | Yes         | Debug with VHE                                                        |
| FEAT_PAN3      | Yes         | Support for SCTL_ELx.EPAN                                             |

**Table 1-4: Implementation status of the Arm®v8.2-A features in the Cortex®-X925 core**

| Feature        | Implemented | Description                                                     |
|----------------|-------------|-----------------------------------------------------------------|
| FEAT_TTCNP     | Yes         | Common not private translations                                 |
| FEAT_XNX       | Yes         | Execute-never control distinction by Exception level at stage 2 |
| FEAT_UAO       | Yes         | Unprivileged Access Override control                            |
| FEAT_PAN2      | Yes         | AT S1E1R and AT S1E1W instruction variants for PAN              |
| FEAT_DPB       | Yes         | DC CVAP instruction                                             |
| FEAT_Debugv8p2 | Yes         | Arm®v8.2-A Debug                                                |
| FEAT_IESB      | Yes         | Implicit Error synchronization event                            |
| FEAT_AA32HPD   | No          | AArch32 Hierarchical permission disables                        |
| FEAT_HPDS2     | Yes         | Hierarchical permission disables in translation tables 2        |
| FEAT_LSMAOC    | No          | Load/Store instruction multiple atomicity and ordering controls |

| Feature       | Implemented  | Description                                                                                                                                            |
|---------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEAT_FP16     | Yes          | Half-precision floating-point data processing                                                                                                          |
| FEAT_LVA      | No           | Large VA support                                                                                                                                       |
| FEAT_LPA      | No           | Large PA and IPA support                                                                                                                               |
| FEAT_VPIPT    | No           | VMID-aware PIPT instruction cache                                                                                                                      |
| FEAT_PCSRv8p2 | Yes          | PC Sample-based profiling version 8.2                                                                                                                  |
| FEAT_RAS      | Yes          | <i>Reliability, Availability, and Serviceability</i> (RAS) Extension version 1.1                                                                       |
| FEAT_SPE      | Yes          | <i>Statistical Profiling Extension</i> (SPE)<br><br>For more information, see <a href="#">21. Statistical Profiling Extension support</a> on page 161. |
| FEAT_SVE      | Yes          | <i>Scalable Vector Extension</i> (SVE)                                                                                                                 |
| FEAT_SHA512   | Yes          | Advanced SIMD SHA512 instructions                                                                                                                      |
| FEAT_SHA3     | Configurable | Advanced SIMD EOR3, RAX1, XAR, and BCAX instructions                                                                                                   |
| FEAT_SM3      |              | Advanced SIMD SM3 instructions                                                                                                                         |
| FEAT_SM4      |              | Advanced SIMD SM4 instructions                                                                                                                         |
| FEAT_DotProd  | Yes          | Advanced SIMD Int8 dot product instructions                                                                                                            |
| FEAT_FHM      | Yes          | Half-precision floating-point FMLAL instructions                                                                                                       |
| FEAT_EVT      | Yes          | Enhanced Virtualization Traps                                                                                                                          |
| FEAT_DPB2     | Yes          | DC CVADP instruction                                                                                                                                   |
| FEAT_BF16     | Yes          | AArch64 BFloat16 instructions                                                                                                                          |
| FEAT_AA32BF16 | No           | AArch32 BFloat16 instructions                                                                                                                          |
| FEAT_I8MM     | Yes          | Int8 Matrix Multiplication                                                                                                                             |
| FEAT_AA32I8MM | No           | AArch32 Int8 Matrix Multiplication                                                                                                                     |
| FEAT_F32MM    | No           | SVE single-precision floating-point matrix multiply instruction                                                                                        |
| FEAT_F64MM    | No           | SVE double-precision floating-point matrix multiply instruction                                                                                        |

**Table 1-5: Implementation status of the Arm®v8.3-A features in the Cortex®-X925 core**

| Feature            | Implemented | Description                                               |
|--------------------|-------------|-----------------------------------------------------------|
| FEAT_PAuth         | Yes         | Pointer authentication                                    |
| FEAT_EPAC          | No          | Enhanced Pointer authentication                           |
| FEAT_PACIMP        | No          | Pointer authentication - IMPLEMENTATION DEFINED algorithm |
| FEAT_PACQARMA5     | No          | Pointer authentication - QARMA5 algorithm                 |
| FEAT_PACQARMA3     | Yes         | Pointer authentication - QARMA3 algorithm                 |
| FEAT_CONSTPACFIELD | Yes         | PAC Algorithm enhancement                                 |
| FEAT_JSCVT         | Yes         | JavaScript FJCVTS conversion instruction                  |
| FEAT_NV            | No          | Nested virtualization                                     |
| FEAT_LRCPC         | Yes         | Load-acquire RCpc instructions                            |
| FEAT_FCMA          | Yes         | Floating-point FCMLA and FCADD instructions               |
| FEAT_CCIDX         | Yes         | Extended cache index                                      |
| FEAT_SPEv1p1       | Yes         | Statistical Profiling Extensions version 1.1              |
| FEAT_DoPD          | Yes         | Debug over Powerdown                                      |

| Feature          | Implemented | Description                                                                                           |
|------------------|-------------|-------------------------------------------------------------------------------------------------------|
| FEAT_PAuth2      | Yes         | Enhancements to pointer authentication                                                                |
| FEAT_FPAC        | Yes         | Faulting on pointer authentication instructions<br><i>Faulting Pointer Authentication Code (FPAC)</i> |
| FEAT_FPACCOMBINE | Yes         | Faulting on combined pointer authentication instructions                                              |

**Table 1-6: Implementation status of the Arm®v8.4-A features in the Cortex®-X925 core**

| Feature          | Implemented | Description                                                                                                                                                                                                                                                                                                                                          |
|------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEAT_SEL2        | Yes         | Secure EL2                                                                                                                                                                                                                                                                                                                                           |
| FEAT_NV2         | No          | Enhanced support for nested virtualization                                                                                                                                                                                                                                                                                                           |
| FEAT_S2FWB       | Yes         | Stage 2 forced write-back                                                                                                                                                                                                                                                                                                                            |
| FEAT_DIT         | Yes         | Data Independent Timing instructions                                                                                                                                                                                                                                                                                                                 |
| FEAT_IDST        | Yes         | ID space trap handling                                                                                                                                                                                                                                                                                                                               |
| FEAT_FlagM       | Yes         | Condition flag manipulation                                                                                                                                                                                                                                                                                                                          |
| FEAT_LSE2        | Yes         | Large System Extensions version 2                                                                                                                                                                                                                                                                                                                    |
| FEAT_LRCPC2      | Yes         | Load-acquire RCpc instructions version 2                                                                                                                                                                                                                                                                                                             |
| FEAT_TLBIOS      | Yes         | TLB invalidate outer-shared instructions                                                                                                                                                                                                                                                                                                             |
| FEAT_TLBIRANGE   | Yes         | TLB range invalidate range instructions                                                                                                                                                                                                                                                                                                              |
| FEAT_TTL         | Yes         | Translation Table Level                                                                                                                                                                                                                                                                                                                              |
| FEAT_BBM         | Yes         | Translation table break before make levels                                                                                                                                                                                                                                                                                                           |
| FEAT_RASv1p1     | Yes         | <i>Reliability, Availability, and Serviceability (RAS)</i> Extension version 1.1<br><br>All extensions up to Arm®v9.0-A at full containment capability with <i>Error Correcting Code (ECC)</i> configured.<br><br>See <a href="#">10. RAS Extension support</a> on page 88 for more information on the implementation of this extension in the core. |
| FEAT_DoubleFault | Yes         | Double Fault Extension                                                                                                                                                                                                                                                                                                                               |
| FEAT_Debugv8p4   | Yes         | Debug relaxations and extensions version 8.4                                                                                                                                                                                                                                                                                                         |
| FEAT_PMUv3p4     | Yes         | PMU extension version 3.4                                                                                                                                                                                                                                                                                                                            |
| FEAT_TRF         | Yes         | Self hosted Trace Extensions                                                                                                                                                                                                                                                                                                                         |
| FEAT_TTST        | Yes         | Small translation tables                                                                                                                                                                                                                                                                                                                             |
| FEAT_AMUv1       | Yes         | Activity Monitors Extension                                                                                                                                                                                                                                                                                                                          |
| FEAT_MPAM        | Yes         | <i>Memory Partitioning and Monitoring (MPAM)</i><br><br>For more information on the <i>Memory System Resource Partitioning and Monitoring (MPAM)</i> Extension, see the <a href="#">Arm® Architecture Reference Manual for A-profile architecture</a> .                                                                                              |

**Table 1-7: Implementation status of the Arm®v8.5-A features in the Cortex®-X925 core**

| Feature      | Implemented | Description                                              |
|--------------|-------------|----------------------------------------------------------|
| FEAT_FlagM2  | Yes         | Condition flag manipulation version 2                    |
| FEAT_FRINTTS | Yes         | FRINT32Z, FRINT32X, FRINT64Z, and FRINT64X instructions  |
| FEAT_ExS     | No          | Disabling context synchronizing exception entry and exit |
| FEAT_GTG     | Yes         | Guest translation granule size                           |



| Feature       | Implemented                                                      | Description                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEAT_BTI      | Yes                                                              | <i>Branch Target Identification</i> (BTI)                                                                                                                                                                                                                                                                                                                                            |
| FEAT_EOPD     | Yes                                                              | Preventing ELO access to halves of address maps                                                                                                                                                                                                                                                                                                                                      |
| FEAT_RNG      | No                                                               | Random number generator                                                                                                                                                                                                                                                                                                                                                              |
| FEAT_RNG_TRAP | No                                                               | Trapping support for RNDR and RNDRRS                                                                                                                                                                                                                                                                                                                                                 |
| FEAT_MTE      | Yes                                                              | <p>Instruction-only Memory Tagging Extension</p> <p>The Cortex®-X925 core always implements the <i>Memory Tagging Extension</i> (MTE) and therefore is compliant with the CHI.E protocol.</p> <p>For information on CHI.E commands inferred by MTE, see the <i>CHI master interface</i> chapter in the <a href="#">Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual</a>.</p> |
| FEAT_MTE2     | Yes                                                              | Full Memory Tagging Extension                                                                                                                                                                                                                                                                                                                                                        |
| FEAT_MTE3     | Configurable                                                     | MTE Asymmetric Fault Handling                                                                                                                                                                                                                                                                                                                                                        |
|               | These features are enabled by setting the BROADCASTMTE pin to 1. |                                                                                                                                                                                                                                                                                                                                                                                      |
| FEAT_PMUv3p5  | Yes                                                              | PMU Extension version 3.5                                                                                                                                                                                                                                                                                                                                                            |

**Table 1-8: Implementation status of the Arm®v8.6-A features in the Cortex®-X925 core**

| Feature       | Implemented | Description                                    |
|---------------|-------------|------------------------------------------------|
| FEAT_ECV      | Yes         | Enhanced counter virtualization                |
| FEAT_FGT      | Yes         | Fine Grain Traps                               |
| FEAT_TWED     | No          | Delayed trapping of WFE                        |
| FEAT_AMUv1p1  | No          | Activity Monitors Extension version 1.1        |
| FEAT_MPAMv0p1 | No          | Memory Partitioning and Monitoring version 0.1 |
| FEAT_MPAMv1p1 | Yes         | Memory Partitioning and Monitoring version 1.1 |
| FEAT_MTPMU    | No          | Multi-threaded PMU Extensions                  |

**Table 1-9: Implementation status of the Arm®v8.7-A features in the Cortex®-X925 core**

| Feature           | Implemented | Description                                                                                               |
|-------------------|-------------|-----------------------------------------------------------------------------------------------------------|
| FEAT_AFP          | Yes         | Alternate floating-point behavior                                                                         |
| FEAT_HCX          | Yes         | Support for the HCRX_EL2 register                                                                         |
| FEAT_LPA2         | No          | Larger physical address for 4KB and 16KB translation granules                                             |
| FEAT_LS64         | No          | Support for 64 byte loads/stores without return                                                           |
| FEAT_LS64_V       | No          | Support for 64-byte stores with return                                                                    |
| FEAT_LS64_ACCDATA | No          | Support for 64-byte ELO stores with return                                                                |
| FEAT_PMUv3p7      | Yes         | <p>Arm®v8.7-A PMU Extensions</p> <p>See <a href="#">17.1 Performance monitors events</a> on page 119.</p> |
| FEAT_RPRES        | No          | Increased precision of Reciprocal Estimate and Reciprocal Square Root Estimate                            |

| Feature      | Implemented | Description                                                                                                              |
|--------------|-------------|--------------------------------------------------------------------------------------------------------------------------|
| FEAT_SPEv1p2 | Yes         | Arm®v8.7-A SPE<br><br>See <a href="#">21. Statistical Profiling Extension support</a> on page 161.                       |
| FEAT_WFXT    | Yes         | WFE and WFI instructions with timeout<br><br>See <a href="#">4.2.1 Wait for Interrupt and Wait for Event</a> on page 39. |
| FEAT_XS      | Yes         | XS attribute                                                                                                             |

**Table 1-10: Implementation status of the Arm®v8.8-A features in the Cortex®-X925 core**

| Feature        | Implemented | Description                                     |
|----------------|-------------|-------------------------------------------------|
| FEAT_CMOW      | No          | Control for cache maintenance permission        |
| FEAT_Debugv8p8 | No          | Debug v8.8                                      |
| FEAT_HBC       | No          | Hinted conditional branch                       |
| FEAT_HPMN0     | Yes         | Setting of MDCR_EL2.HPMN to zero                |
| FEAT_MOPS      | No          | Standardization of memory operations            |
| FEAT_NMI       | No          | Non-maskable Interrupts                         |
| FEAT_PMUv3p8   | No          | Arm®v8.8-A PMU Extensions                       |
| FEAT_PMUv3_TH  | No          | Event counting threshold                        |
| FEAT_SPEv1p3   | No          | Arm®v8.8-A Statistical Profiling Extensions     |
| FEAT_TIDCP1    | No          | EL0 use of IMPLEMENTATION DEFINED functionality |

**Table 1-11: Implementation status of the Arm®v9.0-A features in the Cortex®-X925 core**

| Feature           | Implemented  | Description                                                                                                                     |
|-------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------|
| FEAT_ETE          | Yes          | <i>Embedded Trace Extension (ETE)</i><br><br>See <a href="#">18. Embedded Trace Extension support</a> on page 142.              |
| FEAT_SVE2         | Yes          | <i>Scalable Vector Extension (SVE) version 2</i><br><br>See <a href="#">14. Scalable Vector Extensions support</a> on page 101. |
| FEAT_SVE_AES      | Configurable | SVE AES instructions                                                                                                            |
| FEAT_SVE_PMULL128 |              | SVE PMULL instructions                                                                                                          |
| FEAT_SVE_SHA3     |              | SVE SHA-3 instructions                                                                                                          |
| FEAT_SVE_SM4      |              | SVE SM4 instructions                                                                                                            |
| FEAT_SVE_BitPerm  | Yes          | SVE Bit Permute                                                                                                                 |
| FEAT_TME          | No           | <i>Transactional Memory Extension (TME)</i>                                                                                     |
| FEAT_TRBE         | Yes          | <i>TRace Buffer Extension (TRBE)</i><br><br>See <a href="#">19. Trace Buffer Extension support</a> on page 154.                 |

**Table 1-12: Implementation status of the Arm®v9.1-A features in the Cortex®-X925 core**

| Feature      | Implemented | Description                           |
|--------------|-------------|---------------------------------------|
| FEAT_ETEv1p1 | Yes         | Embedded Trace Extension, version 1.1 |

**Table 1-13: Implementation status of the Arm®v9.2-A features in the Cortex®-X925 core**

| Feature         | Implemented | Description                                                  |
|-----------------|-------------|--------------------------------------------------------------|
| FEAT_BRBE       | No          | Branch Record Buffer Extensions                              |
| FEAT_RME        | No          | Realm Management Extension                                   |
| FEAT_ETEv1p2    | No          | Embedded Trace Extension, version 1.2                        |
| FEAT_SME        | No          | Scalable Matrix Extension                                    |
| FEAT_SME_FA64   | No          | Full A64 support in Streaming mode                           |
| FEAT_SME_F64F64 | No          | Double-precision floating-point outer product instructions   |
| FEAT_SME_I16I64 | No          | 16-bit to 64-bit integer widening outer product instructions |
| FEAT_EBF16      | No          | Enhanced BFloat16                                            |

The following table shows the other standards and specifications that the Cortex®-X925 core supports.

**Table 1-14: Other standards and specifications supported in the Cortex®-X925 core**

| Standard or specification | Version | Description                                                                                                                                                                                                                                                |
|---------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEAT_GICv4p1              | GICv4.1 | <i>Generic Interrupt Controller (GIC)</i><br>See the <a href="#">Arm® Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4</a> for more information.                                                          |
| Debug                     | -       | Arm®v9.2-A architecture implemented with Arm®v8.4-A Debug architecture support and Arm®v8.3-A Debug over powerdown support.<br>See the <a href="#">Arm® Architecture Reference Manual for A-profile architecture</a> for information on this architecture. |
| CoreSight™ architecture   | v3.0    | For more information on CoreSight™ architecture, see the <a href="#">Arm® CoreSight™ Architecture Specification v3.0</a> .                                                                                                                                 |

## Related information

[2.1 Core components](#) on page 30

## 1.5 Test features

The Cortex®-X925 core provides test signals that enable the use of both *Automatic Test Pattern Generation* (ATPG) and *Memory Built-In Self Test* (MBIST) to test the core logic and memory arrays.

The Cortex®-X925 core includes an ATPG test interface that provides signals to control the *Design for Test* (DFT) features of the core. To prevent problems with DFT implementation, you must carefully consider how you use these signals.

Arm also provides MBIST interfaces that enable you to test the RAMs at operational frequency. You can add your own MBIST controllers to automatically generate test patterns and perform result

comparisons. Optionally, you can use your EDA tool to test the physical RAMs directly instead of using the supplied Arm interfaces.

For the list of test signals and information on their usage, see the *Design for Test integration guidelines* chapter in the *Arm® Cortex®-X925 Core Configuration and Integration Manual*.

For the list of external scan control signals, see the *Design for Test integration guidelines* chapter in the *Arm® DynamIQ™ Shared Unit-120 Configuration and Integration Manual*.



The *Arm® Cortex®-X925 Core Configuration and Integration Manual* and *Arm® DynamIQ™ Shared Unit-120 Configuration and Integration Manual* are confidential documents that are available with the appropriate product licenses.

## 1.6 Design tasks

The Cortex®-X925 core is delivered as a synthesizable RTL description in SystemVerilog. Before you can use the Cortex®-X925 core, you must implement, integrate, and program it.

A different party can perform each of the following tasks:

### Implementation

The implementer configures the RTL, adds vendor cells/RAMs, and takes the design through the synthesis and Place and Route (P&R) steps to produce a hard macrocell.

The implementer chooses the options that affect how the RTL source files are rendered. These options can affect the area, maximum frequency, power, and features of the resulting macrocell.

Other components such as *Design for Test* (DFT) structures and, if necessary, power switches can be added to the implementation flow.

### Integration

The integrator connects the macrocell into a *System on Chip* (SoC). This task includes connecting it to a memory system and peripherals.

The integrator configures some features of the core by tying inputs to specific values. These configuration settings affect the start-up behavior before any software configuration is made and can also limit the options available to the software.

### Software programming

The system programmer develops the software to configure and initialize the core and tests the application software.

The programmer configures the core by programming values into registers. The programmed values affect the behavior of the core.

The operation of the final device depends on the build configuration, the configuration inputs, and the software configuration.

See *Functional integration* in the *Arm® DynamIQ™ Shared Unit-120 Configuration and Integration Manual* for signal descriptions.

See *RTL configuration process* in the *Arm® Cortex®-X925 Core Configuration and Integration Manual* and in the *Arm® DynamIQ™ Shared Unit-120 Configuration and Integration Manual* for implementation options.

## 1.7 Product revisions

The following table indicates the main differences in functionality between product revisions.

**Table 1-15: Product revisions**

| Revision | Notes                                                               |
|----------|---------------------------------------------------------------------|
| r0p0     | First release                                                       |
| r0p1     | First release                                                       |
| r0p2     | Errata fixes. Bug fixes and updates for the DSU-120 and CPU bridge. |

Changes in functionality that have an impact on the documentation also appear in [Revision history](#) on page 882.

## 2. Technical overview

The components in the Cortex®-X925 core are designed to make it a high-performance core.

The main blocks include:

- L1 instruction and L1 data memory systems
- L2 memory system
- Register rename
- Instruction decode
- Instruction issue
- Execution pipeline
- *Memory Management Unit* (MMU)
- Trace unit and trace buffer
- *Performance Monitoring Unit* (PMU)
- *Activity Monitoring Unit* (AMU)
- *Generic Interrupt Controller* (GIC) CPU interface

The Cortex®-X925 core interfaces with the *DynamlQ™ Shared Unit-120* through the CPU bridge.

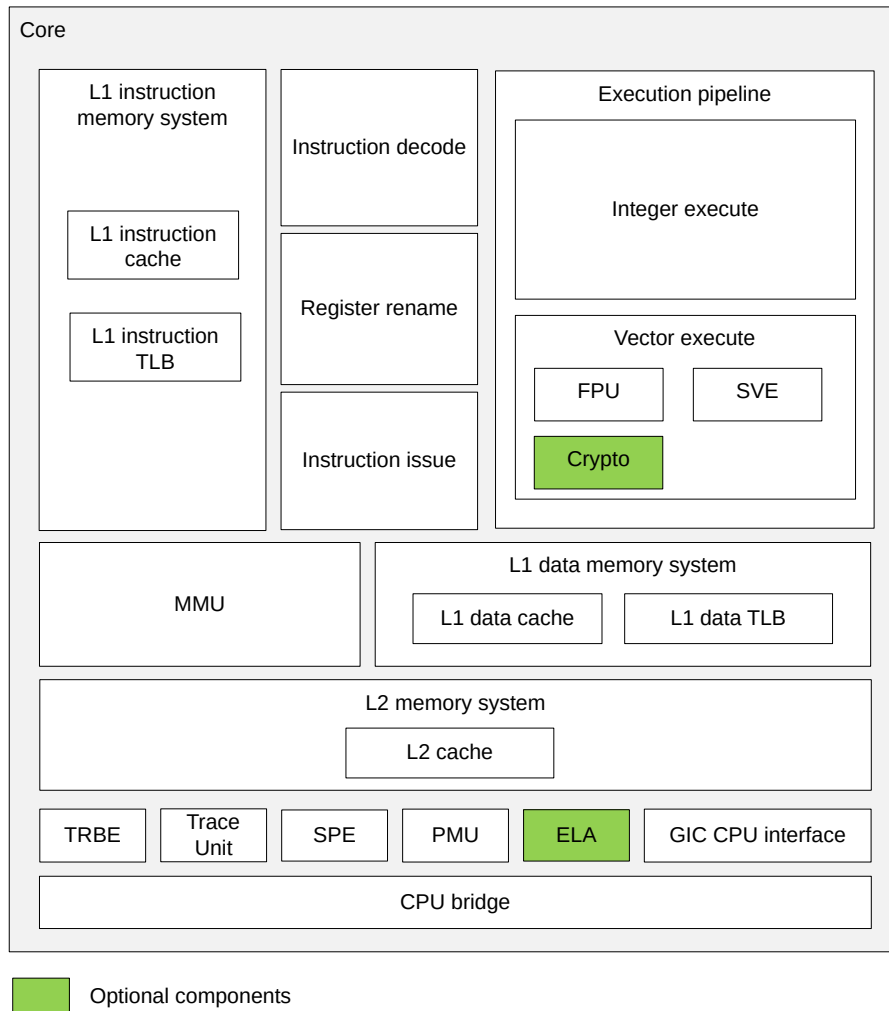
The Cortex®-X925 core implements the Arm®v9.2-A architecture. The Arm®v9.2-A architecture extends the architecture defined in the Arm®v8-A architectures up to Arm®v8.7-A.

The programmer's model and the architecture features implemented, such as the Generic Timer, are compliant with the standards in [1.4 Supported standards and specifications](#) on page 20.

### 2.1 Core components

The Cortex®-X925 core includes components designed to make it a high-performance and low-power product. The Cortex®-X925 core includes a CPU bridge that connects the core to the *DynamlQ™ Shared Unit-120*. The DSU-120 connects the core to an external memory system and the rest of the SoC.

The following figure shows the Cortex®-X925 core components.

**Figure 2-1: Cortex®-X925 core components**

## L1 instruction memory system

The L1 instruction memory system fetches instructions from the instruction cache and delivers the instruction stream to the instruction decode unit.

The L1 instruction memory system includes:

- A 64KB, 4-way set associative L1 instruction cache with 64-byte cache lines.
- A fully associative L1 instruction *Translation Lookaside Buffer* (TLB) with native support for 4KB, 16KB, 64KB, and 2MB page sizes.
- A dynamic branch predictor.

## Instruction decode

The instruction decode unit decodes AArch64 instructions into internal format.

## Register rename

The register rename unit performs register renaming to facilitate out-of-order execution and dispatches decoded instructions to various issue queues.

## Instruction issue

The instruction issue unit controls when the decoded instructions are dispatched to the execution pipelines. It includes issue queues for storing instructions pending dispatch to execution pipelines.

## Integer execute

The integer execution pipeline is part of the overall execution pipeline and includes the integer execute unit that performs arithmetic and logical data processing operations.

## Vector execute

The vector execute unit is part of the execution pipeline and performs Advanced SIMD and floating-point operations (FPU), executes the *Scalable Vector Extension* (SVE) and *Scalable Vector Extension 2* (SVE2) instructions, and can optionally execute the cryptographic instructions (Crypto).

## Advanced SIMD and floating-point support

Advanced SIMD is a media and signal processing architecture that adds instructions primarily for audio, video, 3D graphics, image, and speech processing. The floating-point architecture provides support for single-precision and double-precision floating-point operations.

## Cryptographic Extension

The Cryptographic Extension is optional in the Cortex®-X925 cores. The Cryptographic Extension adds new instructions to the Advanced SIMD and the *Scalable Vector Extension* (SVE) instruction sets that accelerate:

- *Advanced Encryption Standard* (AES) encryption and decryption.
- The *Secure Hash Algorithm* (SHA) functions SHA1, SHA2, SHA3.
  - The SVE2 versions of the SHA3 instructions EOR3, XAR, and BCAX are supported even when CRYPTO support is not configured.
- Armv8.2-SM SM3 hash function and SM4 encryption and decryption instructions.
- Finite field arithmetic that is used in algorithms such as Galois/Counter Mode and Elliptic Curve Cryptography.



The optional Cryptographic Extension is not included in the base product. Arm supplies the Cryptographic Extension under an additional license to the Cortex®-X925 core license.

---

## Scalable Vector Extension

The *Scalable Vector Extension* (SVE) and *Scalable Vector Extension 2* (SVE2) are extensions to the Armv8-A architecture.

SVE complements but does not replace AArch64 Advanced SIMD and floating-point functionality.





The Advanced SIMD architecture, its associated implementations, and supporting software, are also referred to as NEON™ technology.

## L1 data memory system

The L1 data memory system executes load and store instructions and encompasses the L1 data side memory system. It also services memory coherency requests.

The L1 data memory system includes:

- A 64KB, 4-way set associative cache with 64-byte cache lines.
- A fully associative L1 data TLB with native support for 4KB, 16KB and 64KB page sizes and 2MB and 512MB block sizes.

## Memory Management Unit

The *Memory Management Unit* (MMU) provides fine-grained memory system control through a set of virtual-to-physical address mappings and memory attributes that are held in translation tables.

These are saved into the TLB when an address is translated. The TLB entries include global and *Address Space Identifiers* (ASIDs) to prevent context switch TLB invalidations. They also include *Virtual Machine Identifiers* (VMIDs) to prevent TLB invalidations on virtual machine switches by the hypervisor.

## L2 memory system

The L2 memory system includes the L2 cache. The L2 cache is private to the core and can be configured to be 2MB 8-way set associative or 3MB 12-way set associative. The L2 memory system is connected to the DSU-120 through an asynchronous CPU bridge.

## Embedded Trace Extension and Trace Buffer Extension

The Cortex®-X925 core supports a range of debug, test, and trace options including a trace unit and a trace buffer.

The Cortex®-X925 core also includes a ROM table that contains a list of components in the system. Debuggers can use the ROM table to determine which CoreSight components are implemented.

All the debug and trace components of the Cortex®-X925 core are described in this manual. For more information about the *Embedded Logic Analyzer* (ELA), see the [Arm® CoreSight™ ELA-600 Embedded Logic Analyzer Technical Reference Manual](#).

## Statistical Profiling Extension

The Cortex®-X925 core implements the *Statistical Profiling Extension* (SPE) to the Arm®v8.7-A architecture. The SPE provides a statistical view of the performance characteristics of executed instructions that software writers can use to optimize their code for better performance.

## Performance Monitoring Unit

The *Performance Monitoring Unit* (PMU) provides 6 or 31 performance monitors, depending on your configuration. The performance monitors can be configured to gather statistics on the operation of each core and the memory system. The information can be used for debug and code profiling.

## Activity Monitoring Unit

The Cortex®-X925 core implements the Activity Monitors Extension to the Arm®v8.4-A architecture. Activity monitors in the *Activity Monitoring Unit* (AMU) provide useful information for system power management and persistent monitoring.

## GIC CPU interface

The *Generic Interrupt Controller* (GIC) CPU interface, when integrated with an external distributor component, is a resource for supporting and managing interrupts in a cluster system.

## CPU bridge

In a cluster, there is one CPU bridge between each Cortex®-X925 core and the DSU-120.

The CPU bridge controls buffering and synchronization between the core and the DSU-120.

The CPU bridge is asynchronous to allow different frequency, power, and area implementation points for each core. You can configure the CPU bridge to run synchronously without affecting the other interfaces such as debug and trace which are always asynchronous.

## Related information

- 5. [Memory management](#) on page 50
- 6. [L1 instruction memory system](#) on page 59
- 7. [L1 data memory system](#) on page 63
- 8. [L2 memory system](#) on page 68
- 12. [GIC CPU interface](#) on page 96
- 13. [Advanced SIMD and floating-point support](#) on page 100
- 17. [Performance Monitors Extension support](#) on page 119
- 18. [Embedded Trace Extension support](#) on page 142

## 2.2 Interfaces

The *DynamiQ™ Shared Unit-120* manages all Cortex®-X925 core external interfaces to the *System on Chip* (SoC).

See the *Technical overview* chapter in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#) for detailed information on these interfaces.

## 2.3 Programmer's model

The Cortex®-X925 core implements the Arm®v9.2-A architecture. The Arm®v9.2-A architecture extends the architecture defined in the Arm®v8-A architectures up to Arm®v8.7-A. The Cortex®-X925 core supports the AArch64 Execution state at all Exception levels, EL0 to EL3.

For more information about the programmer's model, see [Arm® Architecture Reference Manual for A-profile architecture](#).

### Related information

[1.4 Supported standards and specifications](#) on page 20

### 3. Clocks and resets

To provide dynamic power savings, the Cortex®-X925 core supports hierarchical clock gating. It also supports Warm and Cold resets.

Each Cortex®-X925 core has a single clock domain and receives a single clock input. This clock input is gated by an architectural clock gate in the CPU bridge.

In addition, the Cortex®-X925 core implements extensive clock gating that includes:

- Regional clock gates to various blocks that can gate off portions of the clock tree
- Local clock gates that can gate off individual registers or banks of registers

The Cortex®-X925 core receives the following reset signals from the *DynamIQ™ Shared Unit-120* side of the CPU bridge:

- A Warm reset for all registers in the core except for:
  - Some parts of the Debug logic
  - Some parts of the trace unit logic
  - *Reliability, Availability, and Serviceability* (RAS) logic
  - Performance and Power Management registers. See [4.5 Performance and power management](#) on page 46 for more details on Performance and power management registers.
- A Cold reset for the logic in the core, including the debug logic, trace logic, and RAS logic.

For a complete description of the clock gating and reset scheme of the core, see the following sections in the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#):

- *Clocks and resets*
- *Power and reset control with Power Policy Units*

## 4. Power management

The Cortex®-X925 core provides mechanisms to control both dynamic and static power dissipation.

The dynamic power management includes the following features:

- Hierarchical clock gating
- Per-core *Dynamic Voltage and Frequency Scaling* (DVFS)

The static power management includes the following features:

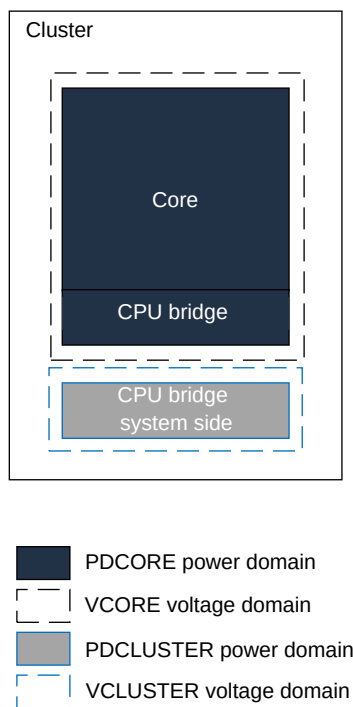
- Powerdown
- Dynamic retention, a low-power mode that retains the register and RAM state

### 4.1 Voltage and power domains

The *DynamlQ™ Shared Unit-120 Power Policy Units* (PPUs) control power management for the Cortex®-X925 core. The core supports one power domain, PDCORE, and one system power domain, PDCLUSTER. Similarly, it supports one core voltage domain, VCORE, and one cluster system voltage domain, VCLUSTER. The power and voltage domains have the same boundaries.

The PDCORE power domain contains all Cortex®-X925 core logic and part of the core asynchronous bridge that belongs to the VCORE domain. The PDCLUSTER power domain contains the part of the CPU bridge that belongs to the VCLUSTER domain.

The following figure shows the Cortex®-X925 core power domain and voltage domain. It also shows the cluster power domain and voltage domain that cover the system side of the CPU bridge.

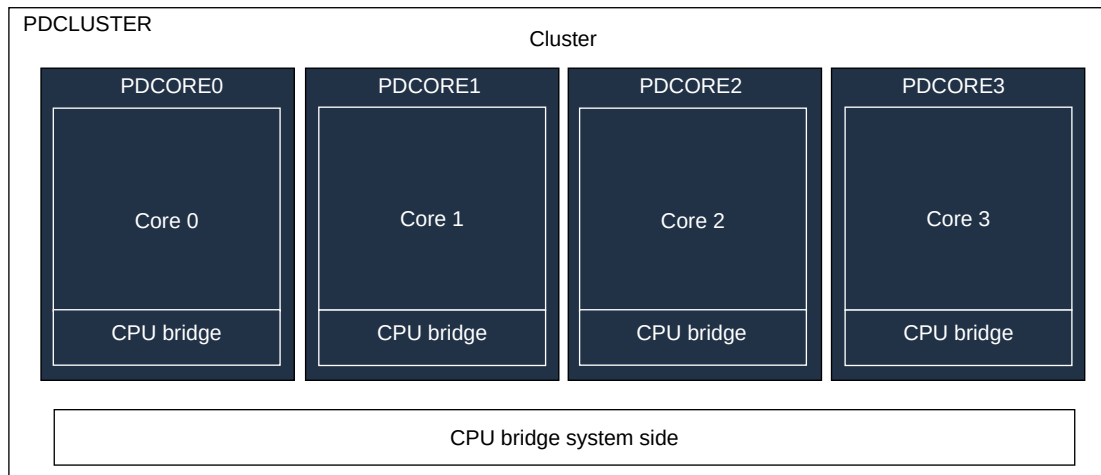
**Figure 4-1: Cortex®-X925 voltage and power domains**

You can tie the VCORE and VCLUSTER voltage domains to the same supply if one of the following is true:

- The core is configured to run synchronously with the DSU-120 sharing the same clock.
- The core is not required to support *Dynamic Voltage and Frequency Scaling* (DVFS).

In a cluster with multiple Cortex®-X925 cores, there is one PDCORE<n> power domain per core, where n is the core instance number. If a core is not present, then the corresponding power domain is not present.

The following figure shows an example of the power domains with four Cortex®-X925 cores in a cluster.

**Figure 4-2: Core power domains in a cluster with four Cortex®-X925 cores**

Clamping cells between power domains are inferred through power intent files rather than instantiated in the RTL. See *Power management* in the *Arm® Cortex®-X925 Core Configuration and Integration Manual* for more information.

For detailed information on the DSU-120 cluster power domains and voltage domains, see *Power management* in the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#).

## 4.2 Architectural clock gating modes

The `WFI` and `WFE` instructions put the core into a low-power mode. These instructions architecturally disable the clock at the top of the clock tree. The core remains fully powered and retains the state.

### 4.2.1 Wait for Interrupt and Wait for Event

*Wait for Interrupt* (WFI) and *Wait for Event* (WFE) are features that put the core in a low-power state by disabling most of the core clocks, while keeping the core powered up. When the core is in WFI or WFE state, the input clock is gated externally to the core at the CPU bridge.

The logic uses a small amount of dynamic power to wake up the core from WFI or WFE low-power state. Other than this power use, the drawn power is reduced to static leakage current only.

When the core executes the `WFI` or `WFE` instruction, it waits for all instructions in the core, including explicit memory accesses, to retire before it enters a low-power state. The `WFI` and `WFE` instructions also ensure that store instructions have updated the cache or have been issued to the L3 memory system.

**Note**

Executing the `WFE` instruction when the event register is set does not cause entry into low-power state, but clears the event register.

The core exits the WFI or WFE state when one of the following events occurs:

- The core detects a reset.
- The core detects one of the architecturally defined WFI or WFE wakeup events.

WFI and WFE wakeup events can include physical and virtual interrupts.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about entering low-power state and wakeup events.

## 4.2.2 Low-power state behavior considerations

You must consider how certain events affect the *Wait for Interrupt* (WFI) and *Wait for Event* (WFE) low-power state behavior of the Cortex®-X925 core.

While the core is in WFI or WFE state, the clocks in the core are temporarily enabled when any of the following events are detected:

- An access on the utility bus interface
- A *Generic Interrupt Controller* (GIC) CPU access
- A debug access through the *Advanced Peripheral Bus* (APB) interface
- A system snoop request that must be serviced by the core L1 data cache or the L2 cache
- A cache or *Translation Lookaside Buffer* (TLB) maintenance operation that must be serviced by the core L1 instruction cache, L1 data cache, L2 cache, or TLB

**Note**

The core does not exit WFI or WFE state when the clocks are temporarily enabled.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about WFI and WFE.



## 4.3 Power control

The *DynamiQ™ Shared Unit-120 Power Policy Units* (PPUs) control all core and cluster power mode transitions.

Each core has its own PPU to control its own core power domain.

In addition, there is a PPU for the cluster.

The PPUs decide and request any change in power mode. The Cortex®-X925 core then performs any actions necessary to reach the requested power mode. For example, the core might gate clocks, clean caches, or disable coherency before it accepts the request.

For more information about the PPUs for the cluster and the cores, see the following sections in the *Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual*:

- *Power management*
- *Power and reset control with Power Policy Units*

### Related information

[A.1.2 IMP\\_CPUPPMCR\\_EL3, Global Performance and Power Management Configuration Register](#) on page 167

[A.1.3 IMP\\_CPUMPMMCR\\_EL3, Global MPMM Control Register](#) on page 169

[B.2.1 CPUPPMCR, Global Performance and Power Management Configuration Register](#) on page 642

[B.2.2 CPUMPMMCR, Global MPMM Control Register](#) on page 644

## 4.4 Core power modes

The Cortex®-X925 core power domain has a defined set of power modes and corresponding legal transitions between these modes. The power mode of each core can be independent of other cores in a cluster.

The *Power Policy Unit* (PPU) of a core manages, at the cluster level, the transitions between the power modes for that core. See *Power management* in the *Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual* for more information.

The following table shows the supported Cortex®-X925 core power modes.

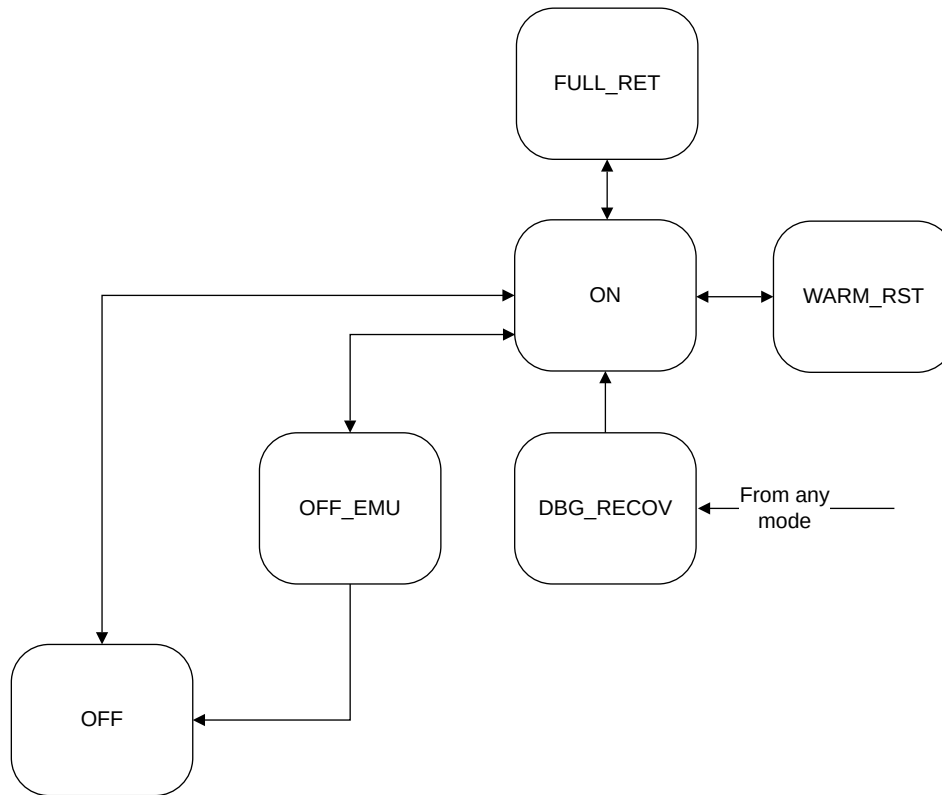


Power modes that are not shown in the following table are not supported and must not occur. Deviating from the legal power modes can lead to **UNPREDICTABLE** results. You must comply with the dynamic power management, and powerup and powerdown sequences described in [4.6 Cortex-X925 core powerup and powerdown sequence](#) on page 48.

**Table 4-1: Cortex®-X925 core power modes**

| Power mode     | Short name | Power state                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| On             | ON         | The core is powered up and active.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Full retention | FULL_RET   | <p>The core is in retention mode. In this mode, only power that is required to retain register and RAM state is available. The core is not operational.</p> <p>A core must be in <i>Wait for Interrupt</i> (WFI) or <i>Wait for Event</i> (WFE) low-power state before it enters this mode.</p>                                                                                                                                                        |
| Off            | OFF        | The core is powered down.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Emulated Off   | OFF_EMU    | <p>Emulated off mode permits you to debug the powerup and powerdown cycle without changing the software.</p> <p>In this mode, the core proceeds through all the powerdown steps, except:</p> <ul style="list-style-type: none"> <li>The clock is not gated and power is not removed when the core is powered down.</li> <li>Only a Warm reset is asserted. The debug logic is preserved in the core and remains accessible by the debugger.</li> </ul> |
| Debug recovery | DBG_RECOV  | <p>The RAM and logic are powered up.</p> <p>This mode is for applying a Warm reset to the DSU-120 DynamIQ™ cluster, while preserving memory and <i>Reliability, Availability, and Serviceability</i> (RAS) registers for debug purposes. Both cache and RAS state are preserved when transitioning from DBG_RECOV to ON.</p> <p><b>Caution:</b><br/>This mode must not be used during normal system operation.</p>                                     |
| Warm reset     | WARM_RST   | A Warm reset resets all state except for the trace logic and the debug and RAS registers.                                                                                                                                                                                                                                                                                                                                                              |

The following figure shows the supported modes for the Cortex®-X925 core power domain and the legal transitions between them.

**Figure 4-3: Cortex®-X925 core power mode transitions****Related information**

[4.2 Architectural clock gating modes](#) on page 39

[4.2.1 Wait for Interrupt and Wait for Event](#) on page 39

[4.4.4 Full retention mode](#) on page 44

**4.4.1 On mode**

In the On power mode, the Cortex®-X925 core is on and fully operational.

The core can be initialized into the On mode. When a transition to the On mode is completed, all caches are accessible and coherent. Other than the normal architectural steps to enable caches, no additional software configuration is required.

**4.4.2 Off mode**

In the Off power mode, power is removed completely from the core and no state is retained.

In Off mode, all core logic and RAMs are off. The domain is inoperable and all core state is lost. The L1 and L2 caches are disabled, cleaned, and invalidated. Also, the core is removed from coherency automatically on transition to Off mode.

A Cold reset can reset the core in this mode.

An attempted external debug access to core debug registers or a utility bus access when the core domain is off returns an error response on the internal debug interface. The error indicates that the core is not available.



The core-specific debug registers in the DebugBlock for *External Debug Over PowerDown* (EDOPD) feature can be accessed while the core is in Off mode.

---

### 4.4.3 Emulated off mode

In Emulated off mode, all core domain logic and RAMs are kept on. All Debug registers must retain their state and be accessible from the external debug interface. All other functional interfaces behave as if the core were in Off mode.

### 4.4.4 Full retention mode

Full retention mode is a dynamic retention mode that is controlled using the *Power Policy Unit* (PPU). On wakeup, full power to the core can be restored and execution can continue.

In Full retention mode, only power that is required to retain register and RAM state is available. The core is in retention state and is non-operational.

The core enters Full retention mode when all of the following conditions are met:

- The retention timer has expired. For more information on setting the retention timer, see [A.2.13 IMP\\_CPUPWRCTLR\\_EL1, CPU Power Control Register](#) on page 216.
- The core is in *Wait for Interrupt* (WFI) or *Wait for Event* (WFE) low-power state.
- The core clock is temporarily disabled for any of the following reasons:
  - L1 snoops or L2 snoops
  - Cache or *Translation Lookaside Buffer* (TLB) maintenance operations
  - Debug or *Generic Interrupt Controller* (GIC) access

The core exits Full retention mode when it detects any of the following events:

- A WFI or WFE wakeup event, as defined in the [Arm® Architecture Reference Manual for A-profile architecture](#).
- An event that requires the core clock to be temporarily enabled without exiting the WFI or WFE low-power state. For example:
  - L1 snoops or L2 snoops
  - Cache or TLB maintenance operations
  - Debug access from the DebugBlock of the *DynamiQ™ Shared Unit-120* (DSU)

- GIC access

## Related information

[4.2.1 Wait for Interrupt and Wait for Event](#) on page 39

### 4.4.5 Debug recovery mode

Debug recovery mode supports debug of external watchdog-triggered reset events, such as watchdog timeout.

By default, the core invalidates its caches when it transitions from Off to On mode. Using Debug recovery mode allows the L1 cache and L2 cache contents that were present before the reset to be observable after the reset. In this mode, the contents of the caches are retained and are not altered on the transition back to the On mode.

In addition to preserving the cache contents, Debug recovery mode supports preserving the *Reliability, Availability, and Serviceability* (RAS) state. A transition to Debug recovery mode is made from any state, which puts the core into a Warm reset state. There is no external mechanism to apply a Warm reset mode other than programming the *DynamiQ™ Shared Unit-120 Power Policy Units* (PPUs).

For more information on the DSU-120 *Power Policy Units* (PPUs), see *The Power Policy Unit* in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#).



Debug recovery is strictly for debug purposes. It must not be used for functional purposes because correct operation of the caches is not guaranteed when entering this mode.

---

Debug recovery mode can occur at any time with no guarantee of the state of the core. A request of this type is accepted immediately, therefore its effects on the core, the DSU-120 DynamiQ™ cluster, or the wider system are **UNPREDICTABLE**, and a wider system reset might be required. In particular, any outstanding memory system transactions at the time of the reset might complete after the reset. The core is not expecting these transactions to complete after a reset, and might cause a system deadlock.

If the system sends a snoop to the DSU-120 DynamiQ™ cluster during Debug recovery mode, depending on the cluster state:

- The snoop might get a response and disturb the contents of the caches.
- The snoop might not get a response and cause a system deadlock.

### 4.4.6 Warm reset mode

A Warm reset resets all state except for the trace logic, debug registers, and *Reliability, Availability, and Serviceability* (RAS) registers.



WARM\_RST mode is strictly for debug purposes.

---

A Warm reset is applied to the Cortex®-X925 core when the core *Power Policy Unit* (PPU) in the *DynamlQ™ Shared Unit-120* is programmed for WARM\_RST mode.

WARM\_RST mode is only expected to be used for resets triggered by a system level issue, such as a watchdog timeout.

WARM\_RST mode can occur at any time with no guarantee of the state of the core. A request to transition to WARM\_RST mode is accepted immediately. Therefore, its effects on the core, the DynamlQ™ cluster, or the wider system are **UNPREDICTABLE**, and a wider system reset might be required. For example, if there were any outstanding memory transactions at the time of the reset, these transactions might complete after the reset. Unless the system interconnect is also reset, the cluster will not expect these transactions to complete after the reset, and a system deadlock might occur.



An alternative method for placing the core into Warm reset is to use the Arm®v8-A Reset Management Register, RMR\_EL3. When the core runs in EL3, it requests a Warm reset of the core if you set the RMR\_EL3.RR bit to 1. If RMR\_EL3.RR is set to 1 before a WFI instruction is executed, then the core will request a Warm reset. The RMR\_EL3.RR controlled Warm reset of the core is independent of the PPU WARM\_RST power mode.

---

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about RMR\_EL3.

## 4.5 Performance and power management

The Cortex®-X925 core implements *Performance and Power Management* (PPM) features that can be used to limit high activity events within the core or trade off efficiency versus peak performance.

The PPM features are:

- *Maximum Power Mitigation Mechanism* (MPMM)
- *Performance Defined Power* (PDP)

## 4.5.1 Maximum Power Mitigation Mechanism

*Maximum Power Mitigation Mechanism* (MPMM) is a power management feature that detects and limits high activity events, specifically high-power load-store events, and vector unit instructions.

If the count of high-activity events exceeds a pre-defined threshold during an evaluation period, MPMM temporarily limits the rate of instruction execution and memory system transactions.

MPMM provides three gears that enable it to limit certain classes of workloads. Each MPMM gear limits workloads at a different level of aggressiveness, where gear 0 produces the most aggressive throttling and gear 2 the least aggressive. The *Activity Monitoring Unit* (AMU) provides metrics for each gear. An external power controller can use these metrics to budget SoC power in the following ways:

- By limiting the number of cores that can execute higher activity workloads
- By switching to a different *Dynamic Voltage and Frequency Scaling* (DVFS) operating point

MPMM is not intended to limit workloads that operate close to typical power levels. The MPMM event detection and limiting are targeted to limit workloads that operate at significantly higher power levels than typical integer workloads.



MPMM must not be relied on as the only electrical safety mechanism. It is essentially a localized assistance mechanism that operates at core level. MPMM is not a substitute for a coarse-grained emergency power reduction scheme, but it does minimize the likelihood of such a scheme being engaged. It is a first line of defense rather than a complete solution.

---

### Related information

[A.1.2 IMP\\_CPUPPMCR\\_EL3, Global Performance and Power Management Configuration Register](#) on page 167

[A.1.3 IMP\\_CPUMPMCR\\_EL3, Global MPMM Control Register](#) on page 169

[B.2.1 CPUPPMCR, Global Performance and Power Management Configuration Register](#) on page 642

[B.2.2 CPUMPMCR, Global MPMM Control Register](#) on page 644

## 4.5.2 Performance Defined Power

*Performance Defined Power* (PDP) is a power management feature that trades off peak performance for a reduced power envelope on general workloads.

The PDP is configured using a level of aggressiveness among three possible values. When the level of aggressiveness is increased, the average workload power is reduced but it causes more performance loss, which varies by workload.

The PDP has an impact on core power reduction. The core power is reduced and the efficiency is increased.

### 4.5.3 Dispatch block

In extreme core thermal or power conditions, you can temporarily halt forward progress of the core without stopping the clock.

A pin is provided on the DSU-120 boundary that can directly be used to force the core to stall for the duration that the pin is asserted. When the core is stalled, the dispatch of new instructions is stopped. However, instructions that have already been dispatched will continue to execute and complete as normal.

## 4.6 Cortex®-X925 core powerup and powerdown sequence

There is no specific sequence to power up the Cortex®-X925 core. To power down the core, you must follow a specific sequence. There are no software steps required to bring a core into coherence after reset.

To power down the Cortex®-X925 core:

1. If required, save the state of the core to system memory to allow for retrieval of the core state during powerup.
2. Disable interrupts to the core.
  - a. Disable the interrupt enable bits in the ICC\_IGRPEN0\_EL1 and ICC\_IGPREN1\_EL1 registers.
  - b. Set the GIC distributor wake-up request for the core using the GICR\_WAKER register.
  - c. Read the GICR\_WAKER register to confirm that the ChildrenAsleep bit indicates that the interface is inactive.
3. Optionally, disable the interrupt outputs from the RAS registers. For more information, see [Managing RAS fault and error interrupts during the core powerdown procedure](#).
4. Set the IMP\_CPUPWRCTLR\_EL1.CORE\_PWRDN\_EN bit to 1 to indicate to the power controller that a powerdown is requested.
5. Execute an `ISB` instruction.
6. Execute a `WFI` instruction. Once the `WFI` instruction is executed, the powerdown sequence cannot be interrupted.

After you have executed the `WFI` and subsequently received a powerdown request from the power controller, the hardware:

- Disables and cleans the core caches
- Removes the core from system coherency

When the IMP\_CPUPWRCTLR\_EL1.CORE\_PWRDN\_EN bit is set, executing a `WFI` instruction automatically masks all interrupts and wakeup events in the core. As a result, applying a reset is the only way to wake up the core from the *Wait for Interrupt* (WFI) state.



### 4.6.1 Managing RAS fault and error interrupts during the core powerdown procedure

After the `WFI` instruction is executed, the power management architecture does not permit interrupting the core software.

The `WFI` instruction is normally the point of no return for powering down the core. However, if a RAS fault or error interrupt is signaled from the core during the power down procedure, this will cause the core to deny the power down request and cause the core to wake up from `WFI`.

Therefore, if the RAS fault and error interrupt outputs remain active during the power down procedure, the software must be designed so that if it wakes up from the power down `WFI`, it can analyze the RAS fault or error and clear the interrupt output before re-executing the `WFI`.

Alternatively, the software could disable the RAS fault and error interrupt outputs before executing the powerdown `WFI` so that the `WFI` is the point of no return for powering down the core. However, this would mean that any detected faults or errors encountered during the powerdown procedure would not be reported and the records of the fault or error would be lost.

## 4.7 Debug over powerdown

The Cortex®-X925 core supports debug over powerdown, which allows a debugger to retain its connection with the core even when powered down. This behavior enables debug to continue through powerdown scenarios rather than having to re-establish a connection each time the core is powered up.

The debug over powerdown logic is part of the DebugBlock in the *DynamiQ™ Shared Unit-120*. The DebugBlock is external to the DSU-120 DynamiQ™ cluster and must remain powered on during the debug over powerdown process.

See *Debug* in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#) for more information.

## 5. Memory management

The *Memory Management Unit* (MMU) translates an input address to an output address.

This translation is based on address mapping and memory attribute information that is available in the Cortex®-X925 core internal registers and translation tables. The MMU also controls memory access permissions, memory ordering, and cache policies for each region of memory.

An address translation from an input address to an output address is described as a stage of address translation. The Cortex®-X925 core can perform:

- Stage 1 translations that translate an input *Virtual Address* (VA) to an output *Physical Address* (PA) or *Intermediate Physical Address* (IPA).
- Stage 2 translations that translate an input IPA to an output PA.
- Combined stage 1 and stage 2 translations that translate an input VA to an IPA, and then translate that IPA to an output PA. The Cortex®-X925 core performs translation table walks for each stage of the translation.

In addition to translating an input address to an output address, a stage of address translation also defines the memory attributes of the output address. With a two-stage translation, the stage 2 translation can modify the attributes that the stage 1 translation defines. A stage of address translation can be disabled or bypassed, and cores can define memory attributes for disabled and bypassed stages of translation.

Each stage of address translation uses address translations and associated memory properties that are held in memory-mapped translation tables. Translation table entries can be cached into a *Translation Lookaside Buffer* (TLB). The translation table entries enable the MMU to provide fine-grained memory system control and to control the table walk hardware.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

### 5.1 Memory Management Unit components

The Cortex®-X925 *Memory Management Unit* (MMU) includes several *Translation Lookaside Buffers* (TLBs), an *MMU Translation Cache* (MMUTC), and a translation table prefetcher.

A TLB is a cache of recently executed page translations within the MMU. The Cortex®-X925 core implements a two-level TLB structure.

A TLB stores all page sizes and is responsible for breaking these down into smaller pages when required for the L1 data or instruction TLB.

The following table describes the MMU components.

**Table 5-1: MMU components**

| Component                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L1 instruction TLB                          | <ul style="list-style-type: none"> <li>Caches entries at the 4KB, 16KB, 64KB, or 2MB granularity of <i>Virtual Address (VA)</i> to <i>Physical Address (PA)</i> mapping only</li> <li>Fully associative</li> <li>128 entries</li> </ul>                                                                                                                                                                                                                                                                                                                                                                    |
| L1 data TLB                                 | <ul style="list-style-type: none"> <li>Caches entries at the 4KB, 16KB, 64KB, 2MB, or 512MB granularity of VA to PA mappings only</li> <li>Fully associative</li> <li>96 entries</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                |
| L1 <i>TRace Buffer Extension (TRBE)</i> TLB | <ul style="list-style-type: none"> <li>VA to PA translations of any page and block size</li> <li>2 entries</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| L2 TLB                                      | <ul style="list-style-type: none"> <li>Shared by instructions and data</li> <li>VA to PA mappings for 4KB, 16KB, 64KB, 2MB, 32MB, 512MB, and 1GB block sizes</li> <li><i>Intermediate Physical Address (IPA)</i> to PA mappings for: <ul style="list-style-type: none"> <li>2MB and 1GB block sizes in a 4KB translation granule</li> <li>32MB block size in a 16KB translation granule</li> <li>512MB block size in a 64KB granule</li> </ul> </li> <li><i>Intermediate PAs</i> (descriptor PAs) obtained during a translation table walk</li> <li>8-way set associative</li> <li>2048 entries</li> </ul> |
| Translation table prefetcher                | <ul style="list-style-type: none"> <li>Detects access to contiguous translation tables and prefetches the next one</li> <li>Can be disabled in the ECTLR register</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                               |

TLB entries contain a global indicator and an *Address Space Identifier (ASID)* to allow context switches without requiring the TLB to be invalidated.

TLB entries contain a *Virtual Machine Identifier (VMID)* to allow virtual machine switches by the hypervisor without requiring the TLB to be invalidated.

## 5.2 Translation Lookaside Buffer entry content

*Translation Lookaside Buffer (TLB)* entries store the context information required to facilitate a match and avoid the need for a TLB clean on a context or virtual machine switch.

Each TLB entry contains:

- A *Virtual Address (VA)*
- A *Physical Address (PA)*
- A set of memory properties that includes type and access permissions

Each TLB entry is associated with either:

- A particular *Address Space Identifier (ASID)*
- A global indicator

Each TLB entry also contains a field to store the *Virtual Machine Identifier* (VMID) in the entry applicable to accesses from EL0 and EL1. The VMID permits hypervisor virtual machine switches without requiring the TLB to be invalidated.

### Related information

[5.4 Translation table walks](#) on page 52

## 5.3 Translation Lookaside Buffer match process

The Armv8-A architecture supports multiple *Virtual Address* (VA) spaces that are translated differently.

Each *Translation Lookaside Buffer* (TLB) entry is associated with a particular translation regime:

- Secure EL3
- Secure EL2
- Secure EL2 and EL0
- Non-secure EL2
- Non-secure EL2 and EL0
- Secure EL1 and EL0
- Non-secure EL1 and EL0

A TLB match entry occurs when the following conditions are met:

- Its VA[48:N], where N is  $\log_2$  of the block size for that translation that is stored in the TLB entry, matches the requested address.
- Entry translation regime matches the current translation regime.
- The *Address Space Identifier* (ASID) matches the current ASID held in the TTBR0\_ELx or TTBR1\_ELx register associated with the target translation regime, or the entry is marked global.
- The *Virtual Machine Identifier* (VMID) matches the current VMID held in the VTTBR\_EL2 register.

The ASID information is used for the purpose of TLB matching for entries using:

- The Secure EL1 and EL0 and Non-secure EL1 and EL0 translation regime
- The Secure EL2 and EL0 and Non-secure EL2 and EL0 translation regime

The VMID information is used for the purpose of TLB matching for entries using:

- The Secure EL1 and EL0 and Non-secure EL1 and EL0 translation regime, when EL2 is enabled.

## 5.4 Translation table walks

When the Cortex®-X925 core generates a memory access, the *Memory Management Unit* (MMU) searches for the requested *Virtual Address* (VA) in the *Translation Lookaside Buffers* (TLBs). If it is not present, then it is a miss and the MMU proceeds by looking up the translation table during a translation table walk.

When the Cortex®-X925 core generates a memory access, the MMU:

1. Performs a lookup for the requested VA, current *Address Space Identifier* (ASID), current *Virtual Machine Identifier* (VMID), and current translation regime in the relevant instruction or data L1 TLB.
2. If there is a miss in the relevant L1 TLB, then the MMU performs a lookup in the L2 TLB for the requested VA, current ASID, current VMID, and translation regime.
3. If there is a miss in the L2 TLB, then the MMU performs a hardware translation table walk.

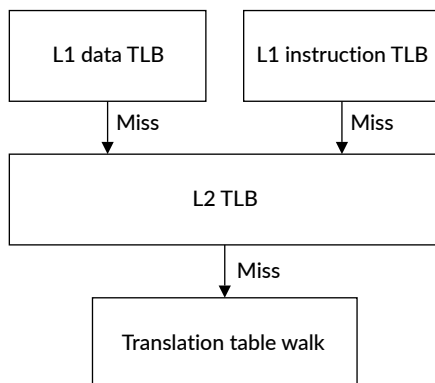
Address translation is performed only when the MMU is enabled. It can also be disabled for a particular translation base register, in which case the MMU returns a Translation fault.

You can program the MMU to make the accesses that are generated by translation table walks cacheable. This means that translation table entries can be cached in the L2 cache, the L3 cache, and external caches.

During a lookup or translation table walk, the access permission bits in the matching translation table entry determine whether the access is permitted. If the permission checks are violated, then the MMU returns a Permission fault. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

The following figure shows the TLB lookup process.

**Figure 5-1: Translation table walks**



In translation table walks, the descriptor is fetched from the L2 or external memory system.

## Related information

6. L1 instruction memory system on page 59

7. L1 data memory system on page 63

8. L2 memory system on page 68

## 5.5 Hardware management of the Access flag and dirty state

The Cortex®-X925 core includes the option to perform hardware updates to the translation tables.

This feature is enabled in TCR\_ELx (where x is 1-3) and VTCR\_EL2. To support hardware management of dirty state, translation table descriptors include the *Dirty Bit Modifier* (DBM) field.

The Cortex®-X925 core supports hardware updates to the Access flag and to dirty state only when the translation tables are held in Inner Write-Back and Outer Write-Back Normal memory regions. If software requests a hardware update in a region that is not Inner Write-Back or Outer Write-Back Normal memory, then the Cortex®-X925 core returns an abort with the following encoding:

- ESR\_ELx.DFSC = 0b110001 for Data Aborts
- ESR\_ELx.IFSC = 0b110001 for Instruction Aborts

## 5.6 Responses

Certain faults and aborts can cause an exception to be taken because of a memory access.

### MMU responses

When one of the following operations is completed, the *Memory Management Unit* (MMU) generates a translation response to the requester:

- An L1 instruction or data *Translation Lookaside Buffer* (TLB) hit
- An L2 TLB hit
- A translation table walk

The responses from the MMU contain the following information:

- The *Physical Address* (PA) that corresponds to the translation
- A set of permissions
- Secure or Non-secure state information
- All the information that is required to report aborts

## MMU aborts

The MMU can detect faults that are related to address translation and can cause exceptions to be taken to the core. Faults can include address size faults, translation faults, access flag faults, and permission faults.

## External aborts

External aborts occur in the memory system and are different from aborts that the MMU detects. Normally, external memory aborts are rare. External aborts are caused by errors that are flagged by the external memory interfaces or are generated because of an uncorrected *Error Correcting Code* (ECC) error in the L1 data cache or L2 cache arrays.

External aborts are reported synchronously when they occur during:

- Translation table walks for instruction fetches, loads, and stores
- Load operations to Inner Write-Back, Outer Write-Back Normal Cacheable memory



The address captured in the *Fault Address Register* (FAR) is the target address of the instruction that generated the synchronous external abort.

---

External aborts are reported asynchronously when they occur during:

- Load operations to all memory locations other than Inner Write-Back, Outer Write-Back Normal memory when the access is not caused by a translation table walk
- Store operations to any memory type
- Cache maintenance, TLB invalidate, and instruction cache invalidate operations
- Atomic operations including `AtomicLd`, `AtomicSt`, `AtomicCAS`, and `AtomicSwap`

Cortex®-X925 takes a synchronous abort on a Normal memory `ldrx` that receives an non-EXOK response from CHI. The abort is asynchronous for Device memory `ldrx`. For `strx`, OK and EXOK responses are expected and do not cause aborts. NDErr and DErr responses for `WriteNoSnp` `Excl=1` cause asynchronous aborts.

## Misprogramming contiguous hints

When there is a descriptor that contains a set CH bit, the input *Virtual Address* (VA) address space must include all contiguous VAs contained in this block.

The VA address space is defined by:

- `TCR_ELx.TxSZ` for stage 1 translations
- `VTCCR_EL2.TOSZ` for stage 2 translations

The Cortex®-X925 core treats such a block as not causing a translation fault and disregards the value of the contiguous bit.

## Conflict aborts

The Cortex®-X925 core does not generate conflict abort exceptions unless configured to do so via IMP\_CPUCTLR\_EL1.[26] (DTLB\_CABT\_EN). Generating conflict abort exceptions this way only applies to Data TLB.

Arm strongly recommends that IMP\_CPUCTLR\_EL1.[26] (DTLB\_CABT\_EN) is not altered from its reset value, which disables conflict aborts.

When a TLB conflict is detected in the L1 TLB or L2 TLB, hardware automatically handles the conflict by invalidating the conflict entries.

## 5.7 Memory behavior and supported memory types

The Cortex®-X925 core supports memory types defined in the Armv8-A architecture.

Device memory types have the following attributes:

### G – Gathering

The capability to gather and merge requests together into a single transaction

### R – Reordering

The capability to reorder transactions

### E – Early Write Acknowledgment

The capability to accept early acknowledgment of write transactions from the interconnect



In the following table, n denotes a "non-" prefix, that is, n means the capability is not allowed.

The following table shows how memory types are supported in the Cortex®-X925 core.

**Table 5-2: Supported device memory types**

| Memory type   | Shareability                 | Inner Cacheability | Outer Cacheability | Notes                    |
|---------------|------------------------------|--------------------|--------------------|--------------------------|
| Device nGnRnE | Outer Shareable              | -                  | -                  | Treated as Device nGnRnE |
| Device nGnRE  | Outer Shareable <sup>1</sup> | -                  | -                  | Treated as Device nGnRE  |
| Device nGRE   | Outer Shareable <sup>1</sup> | -                  | -                  | Treated as Device nGRE   |
| Device GRE    | Outer Shareable <sup>1</sup> | -                  | -                  | Treated as Device GRE    |

<sup>1</sup> Non-cacheable and Device are treated as Outer Shareable. Combinations of Non-cacheable and Write-Through are treated as Non-cacheable, and therefore are Outer Shareable.



| Memory type | Shareability                                                              | Inner Cacheability                         | Outer Cacheability                | Notes                                                                                                                                                  |
|-------------|---------------------------------------------------------------------------|--------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Normal      | Outer Shareable <sup>1</sup>                                              | Non-cacheable                              | Any                               | Treated as Non-cacheable                                                                                                                               |
| Normal      | Outer Shareable <sup>1</sup>                                              | Write-Through Cacheable                    | Any                               | Treated as Non-cacheable                                                                                                                               |
| Normal      | Outer Shareable <sup>1</sup>                                              | Write-Back Cacheable                       | Non-cacheable                     | Treated as Non-cacheable                                                                                                                               |
| Normal      | Outer Shareable <sup>1</sup>                                              | Write-Back Cacheable                       | Write-Through Cacheable           | Treated as Non-cacheable                                                                                                                               |
| Normal      | See <a href="#">Table 5-3: Shareability for Normal memory</a> on page 57. | Write-Back Cacheable (any allocation hint) | Write-Back Cacheable No Allocate  | Treated as Write-Back Read and Write Allocate but the outer cacheability propagated to the <i>DynamiQ™ Shared Unit-120</i> is 0 (No Allocate)          |
| Normal      | See <a href="#">Table 5-3: Shareability for Normal memory</a> on page 57. | Write-Back Cacheable (any allocation hint) | Write-Back Read or Write Allocate | Treated as Write-Back Read and Write Allocate but the outer cacheability propagated to the DSU-120 is 1, therefore upgraded to Write and Read Allocate |

The following table shows how the shareability is treated for certain Normal memory.

**Table 5-3: Shareability for Normal memory**

| Shareability    | Treated as      |
|-----------------|-----------------|
| Non-shareable   | Non-shareable   |
| Outer Shareable | Outer Shareable |
| Inner-Shareable | Outer Shareable |

## 5.8 Page-based hardware attributes

The architecture defines *Page-Based Hardware Attributes* (PBHA) as an optional **IMPLEMENTATION DEFINED** feature. This section describes how the Cortex®-X925 core implements PBHA.

It allows software to set up to four bits in the translation tables, which are then propagated through the memory system with transactions and can be used in the system to control system components. The meaning of the bits is specific to the system design.

For information on how to set and enable the PBHA bits in the translation tables, see the [Arm® Architecture Reference Manual for A-profile architecture](#). When disabled, the PBHA value that is propagated on the bus is 0.

For memory accesses caused by a translation table walk, the IMP\_ATCR\_ELx and IMP\_AVTCR\_EL2 registers control the PBHA values.

### PBHA combination between stage 1 and stage 2 on memory accesses

PBHA should always be considered as an attribute of the physical address.

When stage 1 and stage 2 are enabled:

- If both stage 1 PBHA and stage 2 PBHA are enabled, the final PBHA is stage 2 PBHA.

- If stage 1 PBHA is enabled and stage 2 PBHA is disabled, the final PBHA is stage 1 PBHA.
- If stage 1 PBHA is disabled and stage 2 PBHA is enabled, the final PBHA is stage 2 PBHA.
- If both stage 1 PBHA and stage 2 PBHA are disabled, the final PBHA is defined to 0.

Enable of PBHA has a granularity of 1 bit, so this property is applied independently on each PBHA bit.

### Mismatched aliases

If the same physical address is accessed through more than one virtual address mapping, and the PBHA bits are different in the mappings, then the results are **UNPREDICTABLE**. The PBHA value sent on the bus could be for either mapping.

## 6. L1 instruction memory system

The Cortex®-X925 core L1 instruction memory system fetches instructions and predicts branches. It includes the L1 instruction cache and the L1 instruction *Translation Lookaside Buffer* (TLB).

The L1 instruction memory system provides an instruction stream to the decoder. To increase overall performance and reduce power consumption, the L1 instruction memory system uses dynamic branch prediction and instruction caching.

The following table shows the L1 instruction memory system features.

Table 6-1: L1 instruction memory system features

| Feature              | Description                                                                                                        |
|----------------------|--------------------------------------------------------------------------------------------------------------------|
| L1 instruction cache | 64KB                                                                                                               |
|                      | 4-way set associative                                                                                              |
|                      | <i>Virtually Indexed, Physically Tagged</i> (VIPT) behaving as <i>Physically Indexed, Physically Tagged</i> (PIPT) |
|                      | Always protected with parity                                                                                       |
| Cache line length    | 64 bytes                                                                                                           |
| Cache policy         | <b>L1 I-cache</b><br>Pseudo- <i>Least Recently Used</i> (LRU) cache replacement policy for L1                      |



Note

The L1 instruction TLB also resides in the L1 instruction memory system. However, it is part of the *Memory Management Unit* (MMU) and is described in [5. Memory management](#) on page 50.

### 6.1 L1 instruction cache behavior

The L1 instruction cache is invalidated automatically at reset unless the core power mode is initialized to Debug Recovery.

In Debug Recovery mode, the L1 instruction cache is not functional.

If the L1 instruction cache is disabled, then instruction fetches cannot access any of the instruction cache arrays, except for cache maintenance operations which can execute normally.

If the L1 instruction cache is disabled, then all instruction fetches to cacheable memory are treated as if they were non-cacheable. This treatment means that instruction fetches might not be coherent with caches in other cores, and software must take this into account.



No relationship between cache sets and *Physical Address* (PA) can be assumed. Arm recommends that cache maintenance operations by set/way are used only to invalidate the entire cache.

## Related information

4.4.5 [Debug recovery mode](#) on page 45

## 6.2 L1 instruction cache Speculative memory accesses

Instruction fetches are Speculative and there can be several unresolved branches in the pipeline. A branch instruction or exception in the code stream can cause a pipeline flush, discarding the currently fetched instructions.

On instruction fetches, pages with Device memory type attributes are treated as Non-Cacheable Normal Memory. To prevent instruction fetches, device memory pages must be marked with the translation table descriptor attribute bit *eXecute Never* (XN). The device and code address spaces must be separated in the physical memory map. This separation prevents Speculative fetches to read-sensitive devices when address translation is disabled.

If the L1 instruction cache is enabled and the instruction fetches miss in the L1 instruction cache, then they will look up in L2 and snoop the L1 data cache if those caches are enabled. Instruction fetches are always coherent with data caches, so cache maintenance operations are not required to make stores visible to instruction fetches.

However, the lookup never causes an L1 data cache refill, regardless of the data cache enable status. The line is only allocated in the L2 cache, provided that the L1 instruction cache is enabled. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

## 6.3 Program flow prediction

The Cortex®-X925 core contains program flow prediction hardware, also known as branch prediction. Branch prediction increases overall performance and enhances power efficiency.

Program flow prediction is enabled when the *Memory Management Unit* (MMU) is enabled for the current Exception level. If program flow prediction is disabled, then all taken branches incur a penalty that is associated with cleaning the pipeline. If program flow prediction is enabled, then it predicts whether a conditional or unconditional branch is to be taken, as follows:

- For conditional branches, it predicts whether the branch is to be taken and the address that the branch goes to, known as the branch target address.
- For unconditional branches, it only predicts the branch target address.

Program flow prediction hardware contains the following functionality:

- A *Branch Target Buffer* (BTB) holding the branch target address of previously taken branches
- A *Branch Prediction* (BP) predictor that uses the previous branch history
- The return stack, including nested subroutine return addresses
- A static branch predictor
- An indirect branch predictor

### Predicted and non-predicted instructions

Program flow prediction hardware predicts all branch instructions, and includes:

- Conditional branches
- Unconditional branches
- Return instructions
- Indirect branches

The following instructions are not predicted:

- Exception return instructions (including `ERET`, `ERETAA`, `ERETAB`)
- Supervisor call instructions
- Hypervisor call instructions
- Secure Monitor call instructions

### Return stack

The return stack stores the return address of procedure call instructions. This address should be equal to the value written in the Link Register (X30) by these instructions.

Any of the following instructions causes a return stack push:

- `BL`
- `BLR`
- `BLRAA`
- `BLRAAZ`
- `BLRAB`
- `BLRABZ`

Any of the following instructions cause a return stack pop:

- `RET`
- `RETAA`
- `RETAB`

## 6.4 Instruction Prefetch

Instruction prefetching can boost execution performance by fetching data before it is needed.

### Preload instructions

The Cortex®-X925 core supports the AArch64 prefetch memory instructions, `PRFM PLI`, into the L1 instruction cache or L2 cache.

These instructions signal to the memory system that memory accesses from a specified address are likely to occur soon. The memory system takes actions that aim to reduce the latency of memory accesses when they occur.

The `PRFM PLD` and `PRFM PST` instructions perform preloading in the L1 data cache, L2 cache, or system level cache. `PRFM PLD` and `PRFM PST` instructions translate through the Data TLB.

The `PRFM PLI` instruction performs preloading to the L1 instruction cache and L2 cache. Instruction preloading is performed in the background. `PRFM PLI` instructions translate through the Instruction TLB.

For more information about prefetch memory and preloading caches, see the [Arm® Architecture Reference Manual for A-profile architecture](#).

## 7. L1 data memory system

The Cortex®-X925 L1 data memory system is responsible for executing load and store instructions, as well as specific instructions such as atomics, cache maintenance operations, and memory tagging instructions. It includes the L1 data cache and the L1 data *Translation Lookaside Buffer* (TLB).

The L1 data memory system executes load and store instructions and services memory coherency requests.

The following table shows the L1 data memory system features.

**Table 7-1: L1 data memory system features**

| Feature                                                    | Description                                                                                                                                                                                                             |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L1 data cache                                              | 64KB                                                                                                                                                                                                                    |
|                                                            | 4-way set associative                                                                                                                                                                                                   |
|                                                            | <i>Virtually Indexed, Physically Tagged</i> (VIPT) behaving as <i>Physically Indexed, Physically Tagged</i> (PIPT)                                                                                                      |
|                                                            | Always protected with <i>Error Correcting Code</i> (ECC)                                                                                                                                                                |
| Cache line length                                          | 64 bytes                                                                                                                                                                                                                |
| Cache policy                                               | <i>Re-Reference Interval Prediction</i> (RRIP) replacement policy                                                                                                                                                       |
| Interface with integer execute pipeline and vector execute | <ul style="list-style-type: none"> <li>4×64-bit read paths and 4×64-bit write paths for the integer execute pipeline</li> <li>4×128-bit read paths and 4×128-bit write paths for the vector execute pipeline</li> </ul> |



**Note**

The L1 data TLB also resides in the L1 instruction memory system. However, it is part of the *Memory Management Unit* (MMU) and is described in [5. Memory management](#) on page 50.

### 7.1 L1 data cache behavior

The L1 data cache is invalidated automatically at reset unless the core power mode is initialized to Debug recovery mode.

In Debug recovery mode, the caches are not guaranteed to be functional and should not be enabled.

There is no operation to invalidate the entire data cache. If software requires this function, then it must be constructed by iterating over the cache geometry and executing a series of individual invalidates by set/way instructions. The `dc csw` and `dc isw` instructions perform both a clean and invalidate of the target set/way. The values of `HCR_EL2.SWIO` have no effect. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about `dc csw` and `HCR_EL2`.

## Data Cacheability disabled behavior

If the data Cacheability is disabled, then:

- A new line is not allocated in the L2 or L3 caches as a result of an instruction fetch.
- All load and store instructions to cacheable memory are treated as Non-cacheable.
- Data cache maintenance operations continue to execute normally.

The L1 data and L2 caches cannot be disabled independently. When a core disables the L1 data cache, cacheable memory accesses issued by that core are no longer cached in the L1 or L2 cache.

To maintain data coherency between multiple cores, the Cortex®-X925 core uses the *Modified Exclusive Shared Invalid* (MESI) protocol.



Note

The way that cache indices are determined means that there is no direct relationship between the *Physical Address* (PA) and set number. You cannot use targeted operations that assume a relationship between the PA and set number. To flush the entire cache, you must perform set and way maintenance operations over the number of sets and ways described in CCSIDR\_EL1 for that cache.

## Related information

[4.4.5 Debug recovery mode](#) on page 45

## 7.2 Write streaming mode

The Cortex®-X925 core supports write streaming mode, sometimes referred to as read allocate mode, both for the L1 and the L2 cache.

A cache line is allocated to the L1 and L2 cache on either a read miss or a write miss. However, writing large blocks of data can pollute the cache with unnecessary data. It can also waste power and performance when a linefill is performed only to discard the linefill data because the entire line was subsequently written by the `memset()`. In some situations, cache line allocation on writes is not required. For example, when executing the C standard library `memset()` function to clear a large block of memory to a known value.

To prevent unnecessary cache line allocation, the memory system can detect when the core has written a full cache line before the linefill completes. If this situation is detected on a configurable number of consecutive linefills, then it switches into write streaming mode.

When in write streaming mode, load operations behave as normal, and can still cause linefills. Writes still lookup in the cache, but if they miss then they write out to the L2 or system rather than starting a linefill.



**Note**

More than the specified number of linefills might be observed on the master interface, before the memory system switches to write streaming mode.

The memory system continues in write streaming mode until either:

- It detects a cacheable write burst that is not a full cache line.
- There is a load operation from the same line that is being written.

When a Cortex®-X925 core has switched to write streaming mode, the memory system continues to monitor the write traffic. It signals to the L2, System Level Cache, or DRAM, to go into write streaming mode when it observes a further number of full cache line writes.

The write streaming threshold defines the number of consecutive cache lines that are fully written without being read before store operations stop causing cache allocations. You can configure the write streaming threshold for each cache:

- IMP\_CPUECTLR\_EL1.L2\_WR\_THR configures the L2 write streaming mode threshold
- IMP\_CPUECTLR\_EL1.L3\_WR\_THR configures the L3 write streaming mode threshold
- IMP\_CPUECTLR\_EL1.L4\_WR\_THR configures the L4 write streaming mode threshold
- IMP\_CPUECTLR\_EL1.DRAM\_WR\_THR configures the DRAM write streaming mode threshold

### Related information

[A.2.10 IMP\\_CPUECTLR\\_EL1, CPU Extended Control Register](#) on page 200

## 7.3 Instruction implementation in the L1 data memory system

The Cortex®-X925 core supports the atomic instructions added in the Arm®v8.1-A architecture. Atomic instructions to Cacheable memory can be performed as either near atomics or far atomics, depending on where the cache line containing the data resides.

If an instruction hits in the L1 data cache, then the Cortex®-X925 core tries to perform it as a near atomic. Then, based on system behavior, the core can decide to perform it as a far atomic.

If the operation misses everywhere within the cluster and the interconnect supports far atomics, then the atomic is passed on to the interconnect to perform the operation. If the operation hits anywhere inside the cluster, or if an interconnect does not support atomics, then the L3 memory system performs the atomic operation. If the line is not already there, it allocates the line into the L3 cache.

Therefore if software prefers that the atomic is performed as a near atomic, then precede the atomic instruction with a `PLDW` or `PRFM PSTLKEEP` instruction. Alternatively, CPUECTLR can be programmed such that different types of atomic instructions attempt to execute as a near atomic.

One cache fill is made on an atomic. If the cache line is lost before the atomic operation can be made, then it is sent as a far atomic.

The Cortex®-X925 core supports atomics to Device or Non-cacheable memory, however this relies on the interconnect also supporting atomics. If such an atomic instruction is executed when the interconnect does not support them, then it results in an abort.

## 7.4 Internal exclusive monitor

The Cortex®-X925 core includes an internal exclusive monitor with a 2-state, open and exclusive state machine that manages Load-Exclusive and Store-Exclusive accesses and Clear-Exclusive (CLREX) instructions.

You can use these instructions to construct semaphores, ensuring synchronization between different processes running on the core, and also between different cores that are using the same coherent memory locations for the semaphore. A Load-Exclusive instruction tags a small block of memory for exclusive access. CTR\_ELO defines the size of the tagged blocks as 16 words, one cache line.



A Load-Exclusive or Store-Exclusive instruction in the A64 instruction set is an instruction that has a mnemonic starting with `LDX`, `LDAX`, `STX`, or `STLX`.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information on these instructions.

See [A.5.22 CTR\\_ELO, Cache Type Register](#) on page 348 for more technical reference and register information.

## 7.5 Data prefetching

Data prefetching can boost execution performance by fetching data before it is needed.

### Hardware data prefetcher

The Cortex®-X925 core includes multiple hardware prefetch engines as part of the L1 and L2 caches. These prefetch engines prefetch data into the L1 and L2 caches using a combination of *Virtual Address (VA)* and *Physical Address (PA)*.

The CPUECTLR registers allow control over some aspects of the prefetcher behavior. For more information, see:

- [A.2.10 IMP\\_CPUECTLR\\_EL1, CPU Extended Control Register](#) on page 200
- [A.2.11 IMP\\_CPUECTLR2\\_EL1, CPU Extended Control Register 2](#) on page 210

## Data cache zero

In the Cortex®-X925 core, the *Data Cache Zero by Virtual Address* (DCC ZVA) instruction sets a 64-byte block of memory, which is aligned to 64 bytes, to zero.

For more information, see the [Arm® Architecture Reference Manual for A-profile architecture](#).

# 8. L2 memory system

The Cortex®-X925 core L2 memory system connects the core with the *DynamiQ™ Shared Unit-120* through the CPU bridge. It includes the private L2 cache.

The L2 cache is unified and private to each Cortex®-X925 core in a cluster.

The following table shows the L2 memory system features.

**Table 8-1: L2 memory system features**

| Feature                    | Type                                                                               |
|----------------------------|------------------------------------------------------------------------------------|
| L2 cache                   | 2MB 8-way set associative with 4 banks or 3MB 12-way set associative with 4 banks  |
|                            | <i>Physically Indexed, Physically Tagged (PIPT)</i>                                |
|                            | Always protected with <i>Error Correcting Code (ECC)</i>                           |
| Cache line length          | 64 bytes                                                                           |
| Cache policy               | Dynamic biased cache replacement policy                                            |
| Interface with the DSU-120 | One CHI Issue E compliant interface with 256-bit read and write DAT channel widths |

## 8.1 L2 cache

The integrated L2 cache handles both instruction and data requests from the instruction and data side, as well as translation table walk requests, and snoops from the CHI interconnect.

The L1 instruction cache and L2 cache are weakly inclusive. Instruction fetches that miss in the L1 instruction cache and L2 cache allocate both caches, but the invalidation of the L2 cache does not cause back-invalidates of the L1 instruction cache. The L1 data cache and L2 cache are strictly inclusive. Any data contained in the L1 data cache is also present in the L2 cache. Victimization of L2 cache can cause invalidations of the L1 data cache.

The L2 cache is invalidated automatically at reset unless the core power mode is initialized to Debug Recovery Mode.



Note

The way that cache indices are determined means that there is no direct relationship between the *Physical Address (PA)* and set number. You cannot use targeted operations that assume a relationship between the PA and set number. To flush the entire cache, you must perform set and way maintenance operations over the number of sets and ways described in CCSIDR\_EL1 for that cache. This operation is compliant with the Armv8-A architecture.

### Related information

4.4.5 [Debug recovery mode](#) on page 45

## 8.2 Support for memory types

The Cortex®-X925 core simplifies coherency logic by downgrading some memory types.

Memory that is marked as both Inner Write-Back Cacheable and Outer Write-Back Cacheable is cached in the L1 data cache and the L2 cache.

Memory that is marked as Inner Write-Through is downgraded to Non-cacheable.

Memory that is marked Outer Write-Through or Outer Non-cacheable is downgraded to Non-cacheable, even if the inner attributes are Write-Back Cacheable.

The additional attribute hints are used as follows:

### Allocation hint

Allocation hints help to determine the rules of allocation of newly fetched lines in the system.

### Transient hint

An allocating read to the L1 data cache that has the transient bit set is allocated in the L1 cache. Such reads are marked as most likely to be evicted, according to the L1 eviction policy.

Transient lines evicted from the L2 cache do not allocate downstream caches.

## 8.3 Transaction capabilities

The interface between the Cortex®-X925 core L2 memory system and the *DynamiQ™ Shared Unit-120* provides transaction capabilities for the core.

The following table shows the maximum possible values for read, write, *Distributed Virtual Memory* (DVM) issuing, and snoop capabilities of the Cortex®-X925 core L2 cache.

**Table 8-2: Cortex®-X925 transaction capabilities**

| Attribute                   | Maximum value | Description                                                           |
|-----------------------------|---------------|-----------------------------------------------------------------------|
| Write issuing capability    | 92            | This is the maximum number of outstanding write transactions.         |
| Read issuing capability     | 92            | This is the maximum number of outstanding read transactions.          |
| Snoop acceptance capability | 53            | This is the maximum number of outstanding snoops accepted.            |
| DVM issuing capability      | 92            | This is the maximum number of outstanding DVM operation transactions. |

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for information on the different memory types.

## 9. Direct access to internal memory

The Cortex®-X925 core provides a mechanism to read the internal memory that the L1 caches, L2 cache, and *Translation Lookaside Buffer* (TLB) structures use through **IMPLEMENTATION DEFINED** System registers. When the coherency between the cache data and the system memory data is broken, you can use this mechanism to investigate any issues.



Note

It is not possible to update the contents of the caches or TLB structures.

Direct access to internal memory is available only in EL3. In all other exception levels, executing these instructions results in an Undefined Instruction exception.

You can access the contents of the internal memory using the six read-only (RO) System registers in [Table 9-1: System registers used to access internal memory](#) on page 70. The internal memory is selected by programming the **IMPLEMENTATION DEFINED** RAMINDEX register using the following sys instruction:

```
SYS #6, C15, C0, #0, <Xt>
```

For more information on the RAMINDEX register, see [A.4.1 SYS\\_IMP\\_RAMINDEX, RAM Index](#) on page 291. The data is read from the read-only System registers as shown in the following table.



Note

- All the System registers are read-only (RO) and 64-bits wide
- For the register reset value, see the individual bit resets
- Any access to the data registers returns data
- Click the register name for details on the returned data format

**Table 9-1: System registers used to access internal memory**

| Register name  | Function               | Access | Operation               | Rd Data |
|----------------|------------------------|--------|-------------------------|---------|
| IMP_IDATA0_EL3 | Instruction register 0 | RO     | MRS <Xd>, S3_6_c15_c0_0 | Data    |
| IMP_IDATA1_EL3 | Instruction register 1 | RO     | MRS <Xd>, S3_6_c15_c0_1 | Data    |
| IMP_IDATA2_EL3 | Instruction register 2 | RO     | MRS <Xd>, S3_6_c15_c0_2 | Data    |
| IMP_DDATA0_EL3 | Data register 0        | RO     | MRS <Xd>, S3_6_c15_c1_0 | Data    |
| IMP_DDATA1_EL3 | Data register 1        | RO     | MRS <Xd>, S3_6_c15_c1_1 | Data    |

| Register name  | Function        | Access | Operation               | Rd Data |
|----------------|-----------------|--------|-------------------------|---------|
| IMP_DDATA2_EL3 | Data register 2 | RO     | MRS <Xd>, S3_6_c15_c1_2 | Data    |

## 9.1 L1 cache encodings

Both the L1 data and instruction caches are 4-way set associative.

The size of the configured cache determines the number of sets in each way. The encoding that is used to locate the cache data entry for tag and data memory is set in  $x_n$  in the appropriate `sys` instruction. It is similar for both the tag and data RAM access.

The following tables show the encodings required for locating and selecting a given cache line.

**Table 9-2: Cortex®-X925 L1 instruction cache tag location encoding**

| Bit field of $X_n$ | Description            |
|--------------------|------------------------|
| [31:24]            | RAMID = 0x00           |
| [23:20]            | Reserved               |
| [19:18]            | Way                    |
| [17:14]            | Reserved               |
| [13:6]             | Virtual address [13:6] |
| [5:0]              | Reserved               |

**Table 9-3: Cortex®-X925 L1 instruction cache data location encoding**

| Bit field of $X_n$ | Description            |
|--------------------|------------------------|
| [31:24]            | RAMID = 0x01           |
| [23:20]            | Reserved               |
| [19:18]            | Way                    |
| [17:14]            | Reserved               |
| [13:3]             | Virtual address [13:3] |
| [2:0]              | Reserved               |

**Table 9-4: Cortex®-X925 L1 instruction TLB data location encoding**

| Bit field of $X_n$ | Description       |
|--------------------|-------------------|
| [31:24]            | RAMID = 0x04      |
| [23:8]             | Reserved          |
| [7:0]              | TLB entry (0-127) |

**Table 9-5: Cortex®-X925 L1 data cache tag location encoding**

| Bit field of $X_n$ | Description  |
|--------------------|--------------|
| [31:24]            | RAMID = 0x08 |
| [23:20]            | Reserved     |

| Bit field of Xn | Description                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [19:18]         | Way                                                                                                                                                           |
| [17:16]         | Copy:<br><b>0b00</b><br>Tag RAM associated with Pipe 0<br><b>0b01</b><br>Tag RAM associated with Pipe 1<br><b>0b10</b><br>Reserved<br><b>0b11</b><br>Reserved |
| [15:14]         | Reserved                                                                                                                                                      |
| [13:6]          | Virtual address [13:6]                                                                                                                                        |
| [5:0]           | Reserved                                                                                                                                                      |

**Table 9-6: Cortex®-X925 L1 data cache data location encoding**

| Bit field of Xn | Description            |
|-----------------|------------------------|
| [31:24]         | RAMID = 0x09           |
| [23:20]         | Reserved               |
| [19:18]         | Way                    |
| [17:16]         | BankSel                |
| [15:14]         | Unused                 |
| [13:6]          | Virtual address [13:6] |
| [5:0]           | Reserved               |

**Table 9-7: Cortex®-X925 L1 data TLB location encoding**

| Bit field of Xn | Description      |
|-----------------|------------------|
| [31:24]         | RAMID = 0x0A     |
| [23:6]          | Reserved         |
| [6:0]           | TLB Entry (0-95) |

### 9.1.1 L1 instruction tag RAM returned data

For each register, any access to the L1 instruction tag RAM returns data.

The following tables show the L1 instruction cache tag format for instruction registers.

**Table 9-8: L1 instruction cache tag format for Instruction Register 0**

| Bit field | Description                                    |
|-----------|------------------------------------------------|
| [63:32]   | Reserved                                       |
| [31]      | Non-secure identifier for the physical address |
| [30:3]    | Physical address [39:12]                       |



| Bit field | Description                                                                                                                                                         |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [2:1]     | Instruction state [1:0]<br><br><b>0b00</b><br>Invalid<br><br><b>0b01</b><br>Valid<br><br><b>0b10</b><br><i>Hardware prefetch(HWPRF)</i><br><br><b>0b11</b><br>Valid |
| [0]       | Parity                                                                                                                                                              |

**Table 9-9: L1 instruction cache tag format for Instruction Register 1**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

**Table 9-10: L1 instruction cache tag format for Instruction Register 2**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

## 9.1.2 L1 instruction data RAM returned data

For each register, any access to the L1 instruction data RAM returns data.

The following tables show the L1 instruction cache data format for instruction registers.

**Table 9-11: L1 instruction cache data format for Instruction Register 0**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | Data [63:0] |

**Table 9-12: L1 instruction cache data format for Instruction Register 1**

| Bit field | Description  |
|-----------|--------------|
| [63:12]   | 0            |
| [11:0]    | Data [75:64] |

**Table 9-13: L1 instruction cache data format for Instruction Register 2**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

### 9.1.3 L1 instruction TLB returned data

For each register, any access to the L1 instruction TLB returns data.

The following tables show the L1 instruction TLB format for instruction registers.

**Table 9-14: L1 instruction TLB format for Instruction Register 0**

| Bit field | Description                                                                                                                                                                                                                                                                                                                        |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [63]      | Virtual address [12]                                                                                                                                                                                                                                                                                                               |
| [62:59]   | PBHA [3:0]                                                                                                                                                                                                                                                                                                                         |
| [58]      | TLB attribute                                                                                                                                                                                                                                                                                                                      |
| [57:55]   | Memory attributes:<br><b>0b000</b><br>Device nGnRnE<br><b>0b001</b><br>Device nGnRE<br><b>0b010</b><br>Device nGRE<br><b>0b011</b><br>Device GRE<br><b>0b100</b><br>Non-cacheable<br><b>0b101</b><br>Write-Back No-Allocate<br><b>0b110</b><br>Write-Back Transient<br><b>0b111</b><br>Write-Back Read-Allocate and Write-Allocate |
| [54:52]   | Page size:<br><b>0b000</b><br>4KB<br><b>0b001</b><br>16KB<br><b>0b010</b><br>64KB<br><b>0b100</b><br>2MB<br><b>Other</b><br>Reserved                                                                                                                                                                                               |
| [51]      | Outer-shared                                                                                                                                                                                                                                                                                                                       |
| [50]      | Inner-shared                                                                                                                                                                                                                                                                                                                       |
| [49:42]   | 0                                                                                                                                                                                                                                                                                                                                  |
| [41:40]   | TLB attribute                                                                                                                                                                                                                                                                                                                      |

| Bit field | Description                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [39:24]   | ASID[15:0]                                                                                                                                                                                           |
| [23:8]    | VMID[15:0]                                                                                                                                                                                           |
| [7:5]     | MSID[2:0]:<br><br><b>0b000</b><br>Secure EL1/EL0<br><br><b>0b001</b><br>Secure EL2<br><br><b>0b101</b><br>Secure EL3<br><br><b>0b010</b><br>Non-secure EL1/EL0<br><br><b>0b011</b><br>Non-secure EL2 |
| [4:1]     | TLB attribute                                                                                                                                                                                        |
| [0]       | Valid                                                                                                                                                                                                |

**Table 9-15: L1 instruction TLB format for Instruction Register 1**

| Bit field | Description              |
|-----------|--------------------------|
| [63:36]   | Physical address [39:12] |
| [35:0]    | Virtual address [48:13]  |

**Table 9-16: L1 instruction TLB format for Instruction Register 2**

| Bit field | Description   |
|-----------|---------------|
| [63:4]    | Reserved      |
| [3]       | TLB attribute |
| [2]       | TLB attribute |
| [1]       | Reserved      |
| [0]       | Non-Secure    |

### 9.1.4 L1 data tag RAM returned data

For each register, any access to the L1 data tag RAM returns data.

The following tables show the L1 data cache tag format for data registers.

**Table 9-17: L1 data cache tag format for Data Register 0**

| Bit field | Description                                     |
|-----------|-------------------------------------------------|
| [63:57]   | ECC                                             |
| [56:28]   | Non-secure identifier, physical address [39:12] |
| [27:25]   | Reserved                                        |
| [24]      | Transient/WBNA                                  |

| Bit field | Description                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------|
| [23:20]   | Memory Tagging Extension (MTE) tag poison                                                                                       |
| [19:4]    | MTE tag data                                                                                                                    |
| [3:2]     | MTE tag state:<br><br><b>0b00</b><br>Invalid<br><br><b>0b01</b><br>Shared<br><br><b>0b11</b><br>Dirty state                     |
| [1:0]     | MESI:<br><br><b>0b00</b><br>Invalid<br><br><b>0b01</b><br>Shared<br><br><b>0b10</b><br>Exclusive<br><br><b>0b11</b><br>Modified |

**Table 9-18: L1 data cache tag format for Data Register 1**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

**Table 9-19: L1 data cache tag format for Data Register 2**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

### 9.1.5 L1 data data RAM returned data

For each register, any access to the L1 data data RAM returns data.

The following tables show the L1 data cache data format for data registers.

**Table 9-20: L1 data cache data format for Data Register 0**

| Bit field | Description                        |
|-----------|------------------------------------|
| [63:0]    | word1_data[31:0], word0_data[31:0] |

**Table 9-21: L1 data cache data format for Data Register 1**

| Bit field | Description                        |
|-----------|------------------------------------|
| [63:0]    | word3_data[31:0], word2_data[31:0] |

**Table 9-22: L1 data cache data format for Data Register 2**

| Bit field | Description                                                                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------|
| [63:32]   | 0                                                                                                                          |
| [31:0]    | word3_ecc [6:0], word3_poison, word2_ecc [6:0], word2_poison, word1_ecc [6:0], word1_poison, word0_ecc [6:0], word0_poison |

### 9.1.6 L1 data TLB returned data

For each register, any access to the L1 data TLB returns data.

The following tables show the L1 data TLB format for data registers.

**Table 9-23: L1 data TLB format for Data Register 0**

| Bit field | Description                                                                                                                                                                                                                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [63:62]   | LOR ID [1:0]                                                                                                                                                                                                                                                                                                                             |
| [61]      | LOR match                                                                                                                                                                                                                                                                                                                                |
| [60]      | Outer-shared                                                                                                                                                                                                                                                                                                                             |
| [59]      | Inner-shared                                                                                                                                                                                                                                                                                                                             |
| [58:57]   | S1 translation regime [1:0]                                                                                                                                                                                                                                                                                                              |
| [56:55]   | S2 translation regime [1:0]                                                                                                                                                                                                                                                                                                              |
| [54:52]   | Memory attributes [2:0]:<br><b>0b000</b><br>Device nGnRnE<br><b>0b001</b><br>Device nGnRE<br><b>0b010</b><br>Device nGRE<br><b>0b011</b><br>Device GRE<br><b>0b100</b><br>Non-cacheable<br><b>0b101</b><br>Write-Back No-Allocate<br><b>0b110</b><br>Write-Back Transient<br><b>0b111</b><br>Write-Back Read-Allocate and Write-Allocate |
| [51]      | Outer allocate                                                                                                                                                                                                                                                                                                                           |
| [50]      | S2 <i>Dirty Bit Modifier</i> (DBM) bit                                                                                                                                                                                                                                                                                                   |
| [49]      | S1 DBM bit                                                                                                                                                                                                                                                                                                                               |
| [48]      | TLB coalesced bit                                                                                                                                                                                                                                                                                                                        |
| [47:44]   | Permission bit [3:0]                                                                                                                                                                                                                                                                                                                     |

| Bit field | Description                                                                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [43]      | Device/Non-cacheable HTRAP                                                                                                                                                                                                      |
| [42]      | nG bit                                                                                                                                                                                                                          |
| [41]      | Smash bit                                                                                                                                                                                                                       |
| [40:38]   | Page size [2:0]:<br><br><b>0b000</b><br>4KB<br><b>0b001</b><br>16KB<br><b>0b010</b><br>64KB<br><b>0b011</b><br>Reserved<br><b>0b100</b><br>2MB<br><b>0b101</b><br>Reserved<br><b>0b110</b><br>512MB<br><b>0b111</b><br>Reserved |
| [37:36]   | Security State<br><b>00</b><br>Secure<br><b>01</b><br>Non-secure<br><b>10</b><br>Root<br><b>11</b><br>Realm                                                                                                                     |
| [35:33]   | MSID [1:0]                                                                                                                                                                                                                      |
| [32:17]   | ASID [15:0]                                                                                                                                                                                                                     |
| [16:1]    | VMID [15:0]                                                                                                                                                                                                                     |
| [0]       | Valid                                                                                                                                                                                                                           |

Table 9-24: L1 data TLB format for Data Register 1

| Bit field | Description              |
|-----------|--------------------------|
| [63:37]   | Physical address [38:12] |
| [36:0]    | Virtual address [48:12]  |

Table 9-25: L1 data TLB format for Data Register 2

| Bit field | Description |
|-----------|-------------|
| [8]       | Tagged MTE  |

| Bit field | Description     |
|-----------|-----------------|
| [7:6]     | Reserved        |
| [5]       | FWB override    |
| [4]       | PBHA [3]        |
| [3]       | PBHA [2]        |
| [2]       | PBHA [1]        |
| [1]       | PBHA [0]        |
| [0]       | PA Address [39] |

## 9.2 L2 cache encodings

The 2MB L2 cache is 8-way set associative, and the 3MB L2 cache is 12-way set associative.

The size of the configured cache determines the number of sets in each way. The encoding that is used to locate the cache data entry for tag and data memory is set in `xn` in the appropriate `sxs` instruction. It is similar for both the tag and data RAM access.

The following tables show the encodings required for locating and selecting a given cache line.



Note

The following applies for all tables:

- $\text{Index}[17:8] = \text{XOR}(\text{physical address}[17:8], \text{physical address}[27:18])$
- $\text{Index}[7:6] = \text{XOR}(\text{physical address}[7:6], \text{physical address}[11:10])$

**Table 9-26: Cortex®-X925 L2 cache tag location encoding for 2MB**

| Bit field of Xn | Description                                                    |
|-----------------|----------------------------------------------------------------|
| [31:24]         | RAMID = $0 \times 10$                                          |
| [23:21]         | Reserved                                                       |
| [20:18]         | Way (0-7)                                                      |
| [17:11]         | Index[17:11]                                                   |
| [10:8]          | $\text{XOR}(\text{Index}[10:8], \text{Way}[2:0])$              |
| [7]             | $\text{XOR}(\text{Index}[7], \text{Index}[11])$                |
| [6]             | $\text{XOR}(\text{Index}[6], \text{Index}[10], \text{Way}[2])$ |
| [5:0]           | Reserved                                                       |

**Table 9-27: Cortex®-X925 L2 cache tag location encoding for 3MB**

| Bit field of Xn | Description           |
|-----------------|-----------------------|
| [31:24]         | RAMID = $0 \times 10$ |
| [23:22]         | Reserved              |
| [22:18]         | Way (0-11)            |
| [17:12]         | Index[17:12]          |

| Bit field of Xn | Description                              |
|-----------------|------------------------------------------|
| [11:8]          | XOR(Index[11:8], Way[3:0])               |
| [7:6]           | XOR (Index[7:6], Index[11:10], Way[3:2]) |
| [5:0]           | Reserved                                 |

**Table 9-28: Cortex®-X925 L2 cache data location encoding for 2MB**

| Bit field of Xn | Description                      |
|-----------------|----------------------------------|
| [31:24]         | RAMID = 0x11                     |
| [23:21]         | Reserved                         |
| [20:18]         | Way (0-7)                        |
| [17:11]         | XOR(Index[17:11], 7'b0001000)    |
| [10:8]          | XOR(Index[10:8], Way[2:0])       |
| [7]             | XOR(Index[7], Index[11])         |
| [6]             | XOR(Index[6], Index[10], Way[2]) |
| [5:4]           | Physical Address                 |
| [3:0]           | Reserved                         |

**Table 9-29: Cortex®-X925 L2 cache data location encoding for 3MB**

| Bit field of Xn | Description                             |
|-----------------|-----------------------------------------|
| [31:24]         | RAMID = 0x11                            |
| [23:22]         | Reserved                                |
| [21:18]         | Way (0-11)                              |
| [17:12]         | XOR(Index[17:12], 6'b000100)            |
| [11:8]          | XOR(Index[11:8], Way[3:0])              |
| [7:6]           | XOR(Index[7:6], Index[11:10], Way[3:2]) |
| [5:4]           | Physical Address                        |
| [3:0]           | Reserved                                |

**Table 9-30: Cortex®-X925 L2 TLB location encoding**

| Bit field of Xn | Description       |
|-----------------|-------------------|
| [31:24]         | RAMID = 0x18      |
| [23:21]         | Reserved          |
| [20:18]         | Way (0-7)         |
| [17:8]          | Reserved          |
| [7:0]           | TLB entry (0-255) |

**Table 9-31: Cortex®-X925 2MB and 3MB L2 victim location encoding**

| Bit field of Rd | Description                      |
|-----------------|----------------------------------|
| [31:24]         | RAMID = 0x12                     |
| [23:18]         | Reserved                         |
| [17:8]          | XOR(Index[17:8], 10'b0010000000) |
| [7:6]           | XOR(Index[7:6], Index[11:10])    |



| Bit field of Rd | Description |
|-----------------|-------------|
| [5:0]           | Reserved    |

## 9.2.1 L2 tag RAM returned data

For each register, any access to the L2 tag RAM returns data.

The following tables show the L2 tag cache format for data registers. In the first table:

### For 2MB or 3MB L2 cache

$n=32$ ,  $m=18$

**Table 9-32: L2 tag cache format for Data Register 0**

| Bit field   | Description                                                                                                                                        |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| [63:n+20]   | 0                                                                                                                                                  |
| [n+19:n+13] | ECC                                                                                                                                                |
| [n+12]      | MPAM_PMG                                                                                                                                           |
| [n+11:n+6]  | MPAM_PARTID                                                                                                                                        |
| [n+5]       | MPAM_NS                                                                                                                                            |
| [n+4:n+1]   | PBHA[3:0]                                                                                                                                          |
| [n:11]      | Physical tag [39:m]                                                                                                                                |
| [10]        | Non-secure                                                                                                                                         |
| [9]         | CopyAtHome                                                                                                                                         |
| [8:7]       | Virtual address [13:12]                                                                                                                            |
| [6]         | Shareable                                                                                                                                          |
| [5]         | L1 data cache valid                                                                                                                                |
| [4:3]       | MTE state:<br><br><b>0b00</b><br>Invalid<br><br><b>0b10</b><br>Clean<br><br><b>0b11</b><br>Dirty                                                   |
| [2:0]       | L2 state:<br><br><b>0b101</b><br>UniqueDirty<br><br><b>0b001</b><br>UniqueClean<br><br><b>0bxx11</b><br>SharedClean<br><br><b>0bxx0</b><br>Invalid |

**Table 9-33: L2 tag cache format for Data Register 1**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

**Table 9-34: L2 tag cache format for Data Register 2**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | 0           |

## 9.2.2 L2 data RAM returned data

For each register, any access to the L2 data RAM returns data.

The following tables show the L2 data RAM format for data registers.

**Table 9-35: L2 data RAM format for Data Register 0**

| Bit field | Description |
|-----------|-------------|
| [63:0]    | Data [63:0] |

**Table 9-36: L2 data RAM format for Data Register 1**

| Bit field | Description   |
|-----------|---------------|
| [63:0]    | Data [127:64] |

**Table 9-37: L2 data RAM format for Data Register 2 when L2 ECC granule equals 128**

| Bit field | Description               |
|-----------|---------------------------|
| [63:15]   | 0                         |
| [14:6]    | ECC[8:0] for Data [127:0] |
| [5]       | Poison for Data [127:64]  |
| [4]       | Poison for Data [63:0]    |
| [3:0]     | MTE tags                  |

**Table 9-38: L2 data RAM format for Data Register 2 when L2 ECC granule equals 256**

| Bit field | Description                            |
|-----------|----------------------------------------|
| [63:11]   | 0                                      |
| [10:6]    | ECC[4:0] for Data [255:0] <sup>2</sup> |
| [5]       | Poison for Data [127:64]               |
| [4]       | Poison for Data [63:0]                 |
| [3:0]     | MTE tags                               |

<sup>2</sup> ECC[9:5] for Data[255:0] are contained in the read of the next quad-word.

## 9.2.3 L2 TLB RAM returned data

For each register, any access to the L2 TLB RAM returns data.

The following tables show the L2 TLB format for instruction registers.

L2 TLB format for Instruction Register 0 will have different physical addresses for bits [20:53] if bit[6] is set to 1 or 0.

**Table 9-39: L2 TLB format for Instruction Register 0 if bit[6] is set to 0**

| Bit field | Description                                                                                                                                                                                                                                                                                                                        |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [63]      | Outer Shareable                                                                                                                                                                                                                                                                                                                    |
| [62]      | Inner Shareable                                                                                                                                                                                                                                                                                                                    |
| [61]      | Outer allocate                                                                                                                                                                                                                                                                                                                     |
| [60:58]   | Memory attributes:<br><b>0b000</b><br>Device nGnRnE<br><b>0b001</b><br>Device nGnRE<br><b>0b010</b><br>Device nGRE<br><b>0b011</b><br>Device GRE<br><b>0b100</b><br>Non-cacheable<br><b>0b101</b><br>Write-Back No-Allocate<br><b>0b110</b><br>Write-Back Transient<br><b>0b111</b><br>Write-Back Read-Allocate and Write-Allocate |
| [57:54]   | Reserved                                                                                                                                                                                                                                                                                                                           |
| [53:20]   | Physical address<br><br>When bit[6] is 0: <ul style="list-style-type: none"> <li>[53:26] = PA[39:12]</li> <li>[25:20] = Reserved</li> </ul>                                                                                                                                                                                        |

| Bit field | Description                                                                                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [19:17]   | Page size:<br><b>0b000</b><br>4KB<br><b>0b001</b><br>16KB<br><b>0b010</b><br>64KB<br><b>0b100</b><br>2MB<br><b>0b101</b><br>32MB<br><b>0b110</b><br>512MB<br><b>0b111</b><br>1GB |
| [16:7]    | Reserved                                                                                                                                                                         |
| [6]       | Coalesced entry                                                                                                                                                                  |
| [5:2]     | Valid bits                                                                                                                                                                       |
| [1:0]     | Reserved                                                                                                                                                                         |

**Table 9-40: L2 TLB format for Instruction Register 0 if bit[6] is set to 1**

| Bit field | Description                                                                                                                                                                                                                                                                                                                             |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [63:62]   | Reserved                                                                                                                                                                                                                                                                                                                                |
| [61:20]   | Physical address<br><br>When bit[6] is 1: <ul style="list-style-type: none"> <li>• [53:28] = PA[39:14]</li> <li>• [27:26] = PA[13:12] for page 3 (highest memory address)</li> <li>• [25:24] = PA[13:12]for page 2</li> <li>• [23:22] = PA[13:12]for page 1</li> <li>• [21:20] = PA[13:12]for page 0 (lowest memory address)</li> </ul> |

| Bit field | Description                                                                                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [19:17]   | Page size:<br><b>0b000</b><br>4KB<br><b>0b001</b><br>16KB<br><b>0b010</b><br>64KB<br><b>0b100</b><br>2MB<br><b>0b101</b><br>32MB<br><b>0b110</b><br>512MB<br><b>0b111</b><br>1GB |
| [16:7]    | Reserved                                                                                                                                                                         |
| [6]       | Coalesced entry                                                                                                                                                                  |
| [5:2]     | Valid bits                                                                                                                                                                       |
| [1:0]     | Reserved                                                                                                                                                                         |

**Table 9-41: L2 TLB format for Instruction Register 1**

| Bit field | Description                     |
|-----------|---------------------------------|
| [63:51]   | ASID [12:0]                     |
| [50:47]   | PBHA                            |
| [46]      | Walk cache entry                |
| [45:17]   | Virtual address [48:20]         |
| [16:13]   | Reserved                        |
| [12]      | Non-secure                      |
| [11:1]    | Reserved                        |
| [0]       | nG, indicates a non-global page |

**Table 9-42: L2 TLB format for Instruction Register 2**

| Bit field | Description |
|-----------|-------------|
| [63:22]   | Reserved    |

| Bit field | Description                                                                                                                                                                            |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [21:19]   | MSID [2:0]:<br><br><b>0b000</b><br>Secure EL1<br><br><b>0b001</b><br>Secure EL2<br><br><b>0b010</b><br>Non-secure EL1<br><br><b>0b011</b><br>Non-secure EL2<br><br><b>0b101</b><br>EL3 |
| [18:3]    | VMID [15:0]                                                                                                                                                                            |
| [2:0]     | ASID [15:13]                                                                                                                                                                           |

## 9.2.4 L2 Victim RAM returned data

For each register, any access to the L2 victim RAM returns data.

The following tables show the L2 victim RAM format for data registers.

**Table 9-43: Cortex®-X925 L2 victim format for data register 0 for 2MB L2 cache**

| Bit field of Rd | Description           |
|-----------------|-----------------------|
| [63:56]         | Prefetch              |
| [55:48]         | Data source           |
| [47:40]         | Transient             |
| [39:32]         | Outer allocation hint |
| [31:16]         | Pointer fill counter  |
| [15:0]          | Replacement           |

**Table 9-44: Cortex®-X925 L2 victim format for data register 0 for 3MB L2 cache**

| Bit field of Rd | Description           |
|-----------------|-----------------------|
| [63:60]         | Transient[7:0]        |
| [59:48]         | Outer allocation hint |
| [47:24]         | Pointer fill counter  |
| [23:0]          | Replacement           |

**Table 9-45: Cortex®-X925 L2 victim format for data register 1 for 2MB L2 cache**

| Bit field of Rd | Description |
|-----------------|-------------|
| [63:0]          | 0           |

**Table 9-46: Cortex®-X925 L2 victim format for data register 1 for 3MB L2 cache**

| Bit field of Rd | Description     |
|-----------------|-----------------|
| [63:32]         | 0               |
| [31:20]         | Prefetch        |
| [19:8]          | Source          |
| [7:0]           | Transient[11:4] |

**Table 9-47: Cortex®-X925 L2 victim format for data register 2 for 2MB L2 cache**

| Bit field of Rd | Description |
|-----------------|-------------|
| [63:0]          | 0           |

**Table 9-48: Cortex®-X925 L2 victim format for data register 2 for 3MB L2 cache**

| Bit field of Rd | Description |
|-----------------|-------------|
| [63:0]          | 0           |

## 10. RAS Extension support

The Cortex®-X925 core supports the *Reliability, Availability, and Serviceability* (RAS) Extension, including all extensions up to Arm®v9.2-A.

In particular, the Cortex®-X925 core supports these RAS Extension features:

- *Fault Handling Interrupts* (FHIs)
- *Error Recovery Interrupts* (ERIs)
- Poison attribute on bus transfers
- Cache protection with *Single Error Correct* (SEC) parity on the functional RAMs that only contain clean data. This includes the L1 instruction cache tag, L1 instruction cache data, and the *Memory Management Unit* (MMU) RAMs.
- Cache protection with *Single Error Correct, Double Error Detect* (SECCDED), *Error Correcting Code* (ECC) on the functional RAMs that contain dirty data. This includes the L1 data cache tag, L1 data cache data, L2 cache tag, L2 cache data, and the L2 *Transaction Queue* (TQ) RAMs.
- Error Data Record registers to help software perform recovery actions
- Error injection capabilities to facilitate software and system debug
- The *Error Synchronization Barrier* (ESB) instruction to synchronize unrecoverable errors. When an `esb` instruction is executed, the core ensures that all SError interrupts that are generated by instructions before the `esb` are either taken or deferred. If the core cannot take the interrupt, it records the interrupt in the Deferred Interrupt Status Register DISR\_EL1. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information on DISR\_EL1.

Fault detection features are included in groups within the DSU-120 DynamIQ™ cluster and the Cortex®-X925 core. Each group of fault detection features is referred to as a node. You can access each node by using either the System registers or the utility bus. The following nodes are implemented in the Cortex®-X925 core and the DSU-120 DynamIQ™ cluster:

- Node 0 includes the shared L3 memory system in the *DynamIQ™ Shared Unit-120*.
- Node 1 includes the private L1 memory system, L2 memory system, and the MMU/TLB in the core.

For more information on the architectural RAS Extension and the definition of a node, see the [Arm® Architecture Reference Manual for A-profile architecture](#).

For information on the node that includes the shared L3 memory system, see *RAS extension support* in the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#).



## 10.1 Cache protection behavior

The configuration of the *Reliability, Availability, and Serviceability* (RAS) Extension that is implemented in the Cortex®-X925 core includes cache protection. In this case, the Cortex®-X925 core protects against errors that result in a RAM bitcell holding the incorrect value.

The RAMs in the Cortex®-X925 core have the following capabilities:

### Parity capability

Some RAMs support parity capability, along with *Single-bit Error Correction* (SEC) support. With SEC support, if a parity error is detected, the contents of the RAMs with parity support can invalidate the error entry and refetch the contents from memory.

### ECC capability

Some RAMs support *Error Correcting Code* (ECC) capability, such as the *Single-bit Error Correct*, *Double-bit Error Detect* (SECEDED) scheme.

The following table indicates which protection type is applied to each RAM in the Cortex®-X925 core. The core can progress and remain functionally correct when there is a single-bit error in any RAM.

**Table 10-1: RAM cache protection**

| RAM memory                                          | Parity or ECC support |
|-----------------------------------------------------|-----------------------|
| L1 instruction cache data                           | SEC parity            |
| L1 instruction cache tag                            | SEC parity            |
| L1 data cache data                                  | SECEDED ECC           |
| L1 data cache tag                                   | SECEDED ECC           |
| Memory Management Unit (MMU) Translation Cache (TC) | SEC parity            |
| L2 cache tag                                        | SECEDED ECC           |
| L2 cache data                                       | SECEDED ECC           |
| L2 Transaction Queue (TQ)                           | SECEDED ECC           |

If there are multiple single-bit errors in different RAMs, or within different protection granules within the same RAM, then the core also remains functionally correct.

If there is a double-bit error in a single RAM within the same protection granule, the behavior depends on the RAM:

- For RAMs with SECEDED capability, the core detects, and either reports or defers the error. If the error is in a cache line containing dirty data, then that data might be lost.
- For RAMs with only SEC, the core does not detect a double-bit error, which might cause data corruption.

If there are errors that are three or more bits within the same protection granule, the core might or might not detect the errors. Whether the core detects the errors or not depends on the RAM and the position of the errors within the RAM.

The cache protection feature of the core has a minimal performance impact when no errors are present.

## 10.2 Error containment

The Cortex®-X925 core supports error containment for data errors. This means that detected data errors are not silently propagated. Data errors are deferred using data poisoning to ensure that you are aware of the error. Uncorrectable L1 data cache tag errors and L2 cache tag errors are not containable.

Error containment also implies support for poisoning if there is a double error on an eviction. This ensures that the error of the associated data is reported when it is consumed.

Support for the *Error Synchronization Barrier* (ESB) instruction in the core also allows further isolation of imprecise exceptions that are reported when poisoned data is consumed.

## 10.3 Fault detection and reporting

When the Cortex®-X925 core detects a fault, it raises a *Fault Handling Interrupt* (FHI) exception or an *Error Recovery Interrupt* (ERI) exception through the fault or the error signals. FHIs and ERIs are reflected in the *Reliability, Availability, and Serviceability* (RAS) registers, which are updated in the node that detects the errors.

### Fault handling interrupts

When ERROCTL.R.FI is set, all detected Deferred errors, Uncorrected errors, and overflows of the corrected error counters generate an FHI. When ERROCTL.R.CFI is set, all detected Corrected errors also generate an FHI.

FHIs from core *n* are signaled using nCOREFAULTIRQ[n].

### Error recovery interrupts

When ERROCTL.R.UI is set, all detected Uncorrected errors that are not deferred generate an ERI.

ERIs from core *n* are signaled using nCOREERRIRQ[n].

## 10.4 Error detection and reporting

When the Cortex®-X925 core consumes an error, it raises different exceptions depending on the error type.

The Cortex®-X925 core might raise:

- A *Synchronous External Abort* (SEA)
- An *Asynchronous External Abort* (AEA)

- An *Error Recovery Interrupt* (ERI)

### 10.4.1 Error reporting and performance monitoring

All memory errors detected by *Error Correcting Code* (ECC) or parity errors trigger the MEMORY\_ERROR event.

The *Performance Monitoring Unit* (PMU) counters count the MEMORY\_ERROR event if it is selected and the counter is enabled.

In Secure state, the MEMORY\_ERROR event is counted only if MDCR\_EL3.SPME is asserted. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for a description of MDCR\_EL3.

#### Related information

[17.1 Performance monitors events](#) on page 119

## 10.5 Error injection

Error injection consists of inserting an error in the error detection logic to verify the error handling software.

Error injection uses the error detection and reporting registers to insert errors. The Cortex®-X925 core can inject the following error types:

#### Corrected errors

A *Corrected Error* (CE) is generated for a single-bit *Error Correcting Code* (ECC) error on an L1 data cache access.

#### Deferred errors

A *Deferred Error* (DE) is generated for a double-bit ECC error on eviction of a cache line from the L1 cache to the L2 cache or as a result of a snoop on the L1 cache.

#### Uncontainable errors

An *Uncontainable Error* (UC) is generated for a double-bit ECC error on the L1 tag RAM following an eviction.

An error can be injected immediately or when a 32-bit counter reaches zero. You can control the value of the counter through the Error Pseudo-fault Generation Countdown Register, ERXPFGCDN\_EL1. The value of the counter decrements on a per clock cycle basis. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about ERXPFGCDN\_EL1.



Error injection is a separate source of error within the system and does not create hardware faults.

## 10.6 AArch64 RAS registers

The following summary table provides an overview of all RAS registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 10-2: RAS registers summary**

| Name                          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                         |
|-------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------------|
| <a href="#">ERRIDR_EL1</a>    | 3   | 0   | C5  | C3  | 0   | See individual bit resets. | 64-bit | Error Record ID Register                            |
| <a href="#">ERRSELR_EL1</a>   | 3   | 0   | C5  | C3  | 1   | See individual bit resets. | 64-bit | Error Record Select Register                        |
| <a href="#">ERXFR_EL1</a>     | 3   | 0   | C5  | C4  | 0   | See individual bit resets. | 64-bit | Selected Error Record Feature Register              |
| <a href="#">ERXCTLR_EL1</a>   | 3   | 0   | C5  | C4  | 1   | See individual bit resets. | 64-bit | Selected Error Record Control Register              |
| <a href="#">ERXSTATUS_EL1</a> | 3   | 0   | C5  | C4  | 2   | See individual bit resets. | 64-bit | Selected Error Record Primary Status Register       |
| <a href="#">ERXADDR_EL1</a>   | 3   | 0   | C5  | C4  | 3   | See individual bit resets. | 64-bit | Selected Error Record Address Register              |
| <a href="#">ERXPFGF_EL1</a>   | 3   | 0   | C5  | C4  | 4   | See individual bit resets. | 64-bit | Selected Pseudo-fault Generation Feature Register   |
| <a href="#">ERXPFGCTL_EL1</a> | 3   | 0   | C5  | C4  | 5   | See individual bit resets. | 64-bit | Selected Pseudo-fault Generation Control Register   |
| <a href="#">ERXPFGCDN_EL1</a> | 3   | 0   | C5  | C4  | 6   | See individual bit resets. | 64-bit | Selected Pseudo-fault Generation Countdown Register |
| <a href="#">ERXMISCO_EL1</a>  | 3   | 0   | C5  | C5  | 0   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 0      |
| <a href="#">ERXMISC1_EL1</a>  | 3   | 0   | C5  | C5  | 1   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 1      |
| <a href="#">ERXMISC2_EL1</a>  | 3   | 0   | C5  | C5  | 2   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 2      |
| <a href="#">ERXMISC3_EL1</a>  | 3   | 0   | C5  | C5  | 3   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 3      |
| <a href="#">DISR_EL1</a>      | 3   | 0   | C12 | C1  | 1   | See individual bit resets. | 64-bit | Deferred Interrupt Status Register                  |
| <a href="#">VSESR_EL2</a>     | 3   | 4   | C5  | C2  | 3   | See individual bit resets. | 64-bit | Virtual SError Exception Syndrome Register          |

| Name      | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|-----------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| VDISR_EL2 | 3   | 4   | C12 | C1  | 1   | See individual bit resets. | 64-bit | Virtual Deferred Interrupt Status Register |

## 10.7 External RAS registers

The following summary table provides an overview of all memory-mapped RAS registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 10-3: RAS registers summary**

| Offset | Name                       | Reset                      | Width  | Description                                                 |
|--------|----------------------------|----------------------------|--------|-------------------------------------------------------------|
| 0x0    | <a href="#">ERROFR</a>     | See individual bit resets. | 64-bit | Error Record <n> Feature Register                           |
| 0x8    | <a href="#">ERROCTLR</a>   | See individual bit resets. | 64-bit | Error Record <n> Control Register                           |
| 0x10   | <a href="#">ERROSTATUS</a> | See individual bit resets. | 64-bit | Error Record <n> Primary Status Register                    |
| 0x18   | <a href="#">ERROADDR</a>   | See individual bit resets. | 64-bit | Error Record <n> Address Register                           |
| 0x20   | <a href="#">ERROMISCO</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 0                   |
| 0x28   | <a href="#">ERROMISC1</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 1                   |
| 0x30   | <a href="#">ERROMISC2</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 2                   |
| 0x38   | <a href="#">ERROMISC3</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 3                   |
| 0x800  | <a href="#">ERROPFGF</a>   | See individual bit resets. | 64-bit | Error Record <n> Pseudo-fault Generation Feature Register   |
| 0x808  | <a href="#">ERROPFGCTL</a> | See individual bit resets. | 64-bit | Error Record <n> Pseudo-fault Generation Control Register   |
| 0x810  | <a href="#">ERROPFGCDN</a> | See individual bit resets. | 64-bit | Error Record <n> Pseudo-fault Generation Countdown Register |

# 11. Utility bus

The utility bus provides access to control registers for various system components in the *DynamiQ™ Shared Unit-120* and the cores within the DSU-120 DynamiQ™ cluster. The utility bus is implemented as a 64-bit AMBA AXI5 slave port, and the control registers are memory-mapped onto the utility bus.

The utility bus provides access to the following system functions in the Cortex®-X925 core:

- *Reliability, Availability, and Serviceability* (RAS) registers for the cores
- *Activity Monitor Unit* (AMU) registers in the cores
- *Maximum Power Mitigation Mechanism* (MPMM) registers in the cores



Note

Information about the *Power Policy Unit* (PPU) registers for the cores in the cluster is provided in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#). For all other registers accessed by the utility bus, see *Utility bus* in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#).

## 11.1 Base addresses for system components

Each set of System registers is grouped on separate 64KB page boundaries allowing access to be enforced by a *Memory Management Unit* (MMU).

The following table shows the base addresses for each set of system component registers and what Security state they should be accessed from.



Note

- The base address for each set of registers for the core RAS, AMU, and MPMM registers depend on the core instance number <n>, from 0 to the total number of cores in the cluster minus one.
- In the following table, any address space that is not documented is treated as **RAZ/WI**.
- The base addresses in the following table are the addresses accessed on the utility bus interface. The system interconnect typically maps these addresses into a particular address range based on the system address map. Therefore, software has to add the base address listed here onto the system address range base to get the absolute physical address of a register.

**Table 11-1: Utility bus base addresses for system component registers**

| Base address, n is core instance number | Registers    | Security state | Memory map                                                     |
|-----------------------------------------|--------------|----------------|----------------------------------------------------------------|
| 0x<n>9_0000                             | Core <n> AMU | Both           | <a href="#">B.6 External AMU registers summary</a> on page 738 |
| 0x<n>A_0000                             | Core <n> RAS | Secure         | <a href="#">A.14 AArch64 RAS registers summary</a> on page 547 |

| Base address, n is core instance number | Registers     | Security state | Memory map                                                     |
|-----------------------------------------|---------------|----------------|----------------------------------------------------------------|
| 0x<n>B_0000                             | Core <n> MPMM | Secure         | <a href="#">B.2 External PPM registers summary</a> on page 641 |
| 0x<n>D_0000 - 0x<n>F_0000               | Reserved      | -              | -                                                              |



For more information on utility bus base addresses for system component registers, see the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#).

---

## 12. GIC CPU interface

The *Generic Interrupt Controller* (GIC) supports and controls interrupts. The GIC Distributor connects to the Cortex®-X925 core through a GIC CPU interface. The GIC CPU interface includes registers to mask, identify, and control the state of interrupts that are forwarded to the core.

Each core in a DSU-120 DynamIQ™ cluster has a GIC CPU interface, which connects to a common external Distributor component.

The GICv4.1 architecture implemented in the Cortex®-X925 core supports:

- Two Security states
- Secure virtualization
- *Software-Generated Interrupts* (SGIs)
- Message-based interrupts
- System register access for the CPU interface
- Interrupt masking and prioritization
- Cluster environments, including systems that contain more than eight cores
- Wakeup events in power management environments

The GIC includes interrupt grouping functionality that supports:

- Configuring each interrupt to belong to either Group 0 or Group 1, where Group 0 interrupts are always Secure
- Signaling Group 1 interrupts to the target core using either the IRQ or the FIQ exception request. Group 1 interrupts can be Secure or Non-secure
- Signaling Group 0 interrupts to the target core using the FIQ exception request only
- A unified scheme for handling the priority of Group 0 and Group 1 interrupts

See the [Arm® Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4](#) for more information about interrupt groups.

### 12.1 Disable the GIC CPU interface

The Cortex®-X925 core always includes the *Generic Interrupt Controller* (GIC) CPU interface. However, you can disable it to meet your requirements.

To disable the GIC CPU interface, assert the GICCDISABLE signal HIGH at reset. If you disable it this way, then you can use an external GIC IP to drive the interrupt signals (nFIQ, nIRQ). If the Cortex®-X925 core is not integrated with an external GIC interrupt Distributor component (minimum GICv3 architecture) in the system, then you must disable the GIC CPU interface.

If you disable the GIC CPU interface, then:



- The virtual input signals nVIRQ and nVFIQ and the input signals nIRQ and nFIQ can be driven by an external GIC in the SoC.
- GIC system register access generates **UNDEFINED** instruction exceptions.

**Note**

If you enable the GIC CPU interface, then you must tie off nVIRQ and nVFIQ to HIGH. This is because the GIC CPU interface generates the virtual interrupt signals to the core. The nIRQ and nFIQ signals are controlled by software, therefore there is no requirement to tie them HIGH.

See *Functional integration* in the *Arm® DynamIQ™ Shared Unit-120 Configuration and Integration Manual* for more information on these signals.

## 12.2 AArch64 GIC system registers

The following summary table provides an overview of all GIC system registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 12-1: GIC system registers summary**

| Name           | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                                |
|----------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------------------|
| ICC_PMR_EL1    | 3   | 0   | C4  | C6  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Priority Mask Register                      |
| ICV_PMR_EL1    | 3   | 0   | C4  | C6  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Priority Mask Register              |
| ICC_IARO_EL1   | 3   | 0   | C12 | C8  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Acknowledge Register 0                      |
| ICV_IARO_EL1   | 3   | 0   | C12 | C8  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Acknowledge Register 0              |
| ICC_EOIRO_EL1  | 3   | 0   | C12 | C8  | 1   | See individual bit resets. | 64-bit | Interrupt Controller End Of Interrupt Register 0                           |
| ICV_EOIRO_EL1  | 3   | 0   | C12 | C8  | 1   | See individual bit resets. | 64-bit | Interrupt Controller Virtual End Of Interrupt Register 0                   |
| ICC_HPPIRO_EL1 | 3   | 0   | C12 | C8  | 2   | See individual bit resets. | 64-bit | Interrupt Controller Highest Priority Pending Interrupt Register 0         |
| ICV_HPPIRO_EL1 | 3   | 0   | C12 | C8  | 2   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Highest Priority Pending Interrupt Register 0 |
| ICV_BPRO_EL1   | 3   | 0   | C12 | C8  | 3   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Binary Point Register 0                       |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                                |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------------------|
| ICC_AP0R0_EL1   | 3   | 0   | C12 | C8  | 4   | See individual bit resets. | 64-bit | Interrupt Controller Active Priorities Group 0 Registers                   |
| ICV_AP0R0_EL1   | 3   | 0   | C12 | C8  | 4   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Active Priorities Group 0 Registers           |
| ICC_AP1R0_EL1   | 3   | 0   | C12 | C9  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Active Priorities Group 1 Registers                   |
| ICV_AP1R0_EL1   | 3   | 0   | C12 | C9  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Active Priorities Group 1 Registers           |
| ICC_DIR_EL1     | 3   | 0   | C12 | C11 | 1   | See individual bit resets. | 64-bit | Interrupt Controller Deactivate Interrupt Register                         |
| ICV_DIR_EL1     | 3   | 0   | C12 | C11 | 1   | See individual bit resets. | 64-bit | Interrupt Controller Deactivate Virtual Interrupt Register                 |
| ICC_RPR_EL1     | 3   | 0   | C12 | C11 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Running Priority Register                             |
| ICV_RPR_EL1     | 3   | 0   | C12 | C11 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Running Priority Register                     |
| ICC_SGI1R_EL1   | 3   | 0   | C12 | C11 | 5   | See individual bit resets. | 64-bit | Interrupt Controller Software Generated Interrupt Group 1 Register         |
| ICC_ASIG1R_EL1  | 3   | 0   | C12 | C11 | 6   | See individual bit resets. | 64-bit | Interrupt Controller Alias Software Generated Interrupt Group 1 Register   |
| ICC_SGI0R_EL1   | 3   | 0   | C12 | C11 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Software Generated Interrupt Group 0 Register         |
| ICC_IAR1_EL1    | 3   | 0   | C12 | C12 | 0   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Acknowledge Register 1                      |
| ICV_IAR1_EL1    | 3   | 0   | C12 | C12 | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Acknowledge Register 1              |
| ICC_EOIR1_EL1   | 3   | 0   | C12 | C12 | 1   | See individual bit resets. | 64-bit | Interrupt Controller End Of Interrupt Register 1                           |
| ICV_EOIR1_EL1   | 3   | 0   | C12 | C12 | 1   | See individual bit resets. | 64-bit | Interrupt Controller Virtual End Of Interrupt Register 1                   |
| ICC_HPPIR1_EL1  | 3   | 0   | C12 | C12 | 2   | See individual bit resets. | 64-bit | Interrupt Controller Highest Priority Pending Interrupt Register 1         |
| ICV_HPPIR1_EL1  | 3   | 0   | C12 | C12 | 2   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Highest Priority Pending Interrupt Register 1 |
| ICC_BPR1_EL1    | 3   | 0   | C12 | C12 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Binary Point Register 1                               |
| ICV_BPR1_EL1    | 3   | 0   | C12 | C12 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Binary Point Register 1                       |
| ICC_CTLR_EL1    | 3   | 0   | C12 | C12 | 4   | See individual bit resets. | 64-bit | Interrupt Controller Control Register (EL1)                                |
| ICV_CTLR_EL1    | 3   | 0   | C12 | C12 | 4   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Control Register                              |
| ICC_SRE_EL1     | 3   | 0   | C12 | C12 | 5   | See individual bit resets. | 64-bit | Interrupt Controller System Register Enable Register (EL1)                 |
| ICC_IGRPEN0_EL1 | 3   | 0   | C12 | C12 | 6   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Group 0 Enable Register                     |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                    |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------|
| ICV_IGRPEN0_EL1 | 3   | 0   | C12 | C12 | 6   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Group 0 Enable Register |
| ICC_IGRPEN1_EL1 | 3   | 0   | C12 | C12 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Group 1 Enable Register         |
| ICV_IGRPEN1_EL1 | 3   | 0   | C12 | C12 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Group 1 Enable Register |
| ICH_APOR0_EL2   | 3   | 4   | C12 | C8  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Hyp Active Priorities Group 0 Registers   |
| ICH_AP1R0_EL2   | 3   | 4   | C12 | C9  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Hyp Active Priorities Group 1 Registers   |
| ICC_SRE_EL2     | 3   | 4   | C12 | C9  | 5   | See individual bit resets. | 64-bit | Interrupt Controller System Register Enable Register (EL2)     |
| ICH_HCR_EL2     | 3   | 4   | C12 | C11 | 0   | See individual bit resets. | 64-bit | Interrupt Controller Hyp Control Register                      |
| ICH_VTR_EL2     | 3   | 4   | C12 | C11 | 1   | See individual bit resets. | 64-bit | Interrupt Controller VGIC Type Register                        |
| ICH_MISR_EL2    | 3   | 4   | C12 | C11 | 2   | See individual bit resets. | 64-bit | Interrupt Controller Maintenance Interrupt State Register      |
| ICH_EISR_EL2    | 3   | 4   | C12 | C11 | 3   | See individual bit resets. | 64-bit | Interrupt Controller End of Interrupt Status Register          |
| ICH_ELRSR_EL2   | 3   | 4   | C12 | C11 | 5   | See individual bit resets. | 64-bit | Interrupt Controller Empty List Register Status Register       |
| ICH_VMCR_EL2    | 3   | 4   | C12 | C11 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Machine Control Register          |
| ICH_LR0_EL2     | 3   | 4   | C12 | C12 | 0   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                            |
| ICH_LR1_EL2     | 3   | 4   | C12 | C12 | 1   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                            |
| ICH_LR2_EL2     | 3   | 4   | C12 | C12 | 2   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                            |
| ICH_LR3_EL2     | 3   | 4   | C12 | C12 | 3   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                            |
| ICC_CTLR_EL3    | 3   | 6   | C12 | C12 | 4   | See individual bit resets. | 64-bit | Interrupt Controller Control Register (EL3)                    |
| ICC_SRE_EL3     | 3   | 6   | C12 | C12 | 5   | See individual bit resets. | 64-bit | Interrupt Controller System Register Enable Register (EL3)     |
| ICC_IGRPEN1_EL3 | 3   | 6   | C12 | C12 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Group 1 Enable Register (EL3)   |

## 13. Advanced SIMD and floating-point support

The Cortex®-X925 core supports the Advanced *Single Instruction Multiple Data* (SIMD) and scalar floating-point instructions in the A64 instruction set without floating-point exception trapping.

The Cortex®-X925 core floating-point implementation includes features up to Arm®v9.2-A. BFloat16 floating-point and Int8 matrix multiplication are part of these supported features.

The Cortex®-X925 core implements all operations in hardware with support for all combinations of:

- Rounding modes
- Flush-to-zero
- Default *Not a Number* (NaN) modes

The Cortex®-X925 core supports *Alternate Floating Point* behavior (FEAT\_AFP), as part of Arm®v8.7-A and Arm®v9.2-A.

## 14. Scalable Vector Extensions support

The Cortex®-X925 core supports the *Scalable Vector Extension* (SVE) and the *Scalable Vector Extension 2* (SVE2). SVE and SVE2 are intended to complement, not replace, AArch64 Advanced *Single Instruction Multiple Data* (SIMD) and floating-point functionality.

SVE is an optional extension introduced by the Armv8.2 architecture. The key features that SVE provides are:

- Predication
- Gather-load and scatter-store
- Software-managed speculative vectorization

The Cortex®-X925 core implements a scalable vector length of 128 bits.

All the features and additions that SVE and SVE2 introduce are described in the [Arm® Architecture Reference Manual for A-profile architecture](#).

# 15. System control

The system registers control and provide status information for the functions that the core implements.

The main functions of the system registers are:

- System performance monitoring
- Cache configuration and management
- Overall system control and configuration
- *Memory Management Unit* (MMU) configuration and management
- *Generic Interrupt Controller* (GIC) configuration and management

The system registers are accessible in AArch64 Execution state at EL0 to EL3. Some of the system registers are accessible through the external debug interface or utility bus interface.

## 15.1 AArch64 Generic System Control registers

The following summary table provides an overview of all Generic System Control registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 15-1: Generic System Control registers summary**

| Name                      | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                             |
|---------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------|
| <a href="#">ACTLR_EL1</a> | 3   | 0   | C1  | C0  | 1   | See individual bit resets. | 64-bit | Auxiliary Control Register (EL1)        |
| RGSR_EL1                  | 3   | 0   | C1  | C0  | 5   | See individual bit resets. | 64-bit | Random Allocation Tag Seed Register.    |
| GCR_EL1                   | 3   | 0   | C1  | C0  | 6   | See individual bit resets. | 64-bit | Tag Control Register.                   |
| TTBR0_EL1                 | 3   | 0   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Translation Table Base Register 0 (EL1) |
| TTBR1_EL1                 | 3   | 0   | C2  | C0  | 1   | See individual bit resets. | 64-bit | Translation Table Base Register 1 (EL1) |
| TCR_EL1                   | 3   | 0   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Translation Control Register (EL1)      |

| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                 |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------------|
| APIAKeyLo_EL1 | 3   | 0   | C2  | C1  | 0   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Instruction (bits[63:0])   |
| APIAKeyHi_EL1 | 3   | 0   | C2  | C1  | 1   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Instruction (bits[127:64]) |
| APIBKeyLo_EL1 | 3   | 0   | C2  | C1  | 2   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Instruction (bits[63:0])   |
| APIBKeyHi_EL1 | 3   | 0   | C2  | C1  | 3   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Instruction (bits[127:64]) |
| APDAKeyLo_EL1 | 3   | 0   | C2  | C2  | 0   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Data (bits[63:0])          |
| APDAKeyHi_EL1 | 3   | 0   | C2  | C2  | 1   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Data (bits[127:64])        |
| APDBKeyLo_EL1 | 3   | 0   | C2  | C2  | 2   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Data (bits[63:0])          |
| APDBKeyHi_EL1 | 3   | 0   | C2  | C2  | 3   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Data (bits[127:64])        |
| APGAKeyLo_EL1 | 3   | 0   | C2  | C3  | 0   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Code (bits[63:0])          |
| APGAKeyHi_EL1 | 3   | 0   | C2  | C3  | 1   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Code (bits[127:64])        |
| SPSel         | 3   | 0   | C4  | C2  | 0   | See individual bit resets. | 64-bit | Stack Pointer Select                                        |
| CurrentEL     | 3   | 0   | C4  | C2  | 2   | See individual bit resets. | 64-bit | Current Exception Level                                     |
| PAN           | 3   | 0   | C4  | C2  | 3   | See individual bit resets. | 64-bit | Privileged Access Never                                     |
| UAO           | 3   | 0   | C4  | C2  | 4   | See individual bit resets. | 64-bit | User Access Override                                        |
| AFSR0_EL1     | 3   | 0   | C5  | C1  | 0   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 0 (EL1)                     |
| AFSR1_EL1     | 3   | 0   | C5  | C1  | 1   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 1 (EL1)                     |
| ESR_EL1       | 3   | 0   | C5  | C2  | 0   | See individual bit resets. | 64-bit | Exception Syndrome Register (EL1)                           |
| TFSR_EL1      | 3   | 0   | C5  | C6  | 0   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL1)                             |
| TFSRE0_EL1    | 3   | 0   | C5  | C6  | 1   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL0).                            |
| FAR_EL1       | 3   | 0   | C6  | C0  | 0   | See individual bit resets. | 64-bit | Fault Address Register (EL1)                                |
| PAR_EL1       | 3   | 0   | C7  | C4  | 0   | See individual bit resets. | 64-bit | Physical Address Register                                   |
| MAIR_EL1      | 3   | 0   | C10 | C2  | 0   | See individual bit resets. | 64-bit | Memory Attribute Indirection Register (EL1)                 |
| AMAIR_EL1     | 3   | 0   | C10 | C3  | 0   | See individual bit resets. | 64-bit | Auxiliary Memory Attribute Indirection Register (EL1)       |

| Name                   | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                      |
|------------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------|
| LORSA_EL1              | 3   | 0   | C10 | C4  | 0   | See individual bit resets. | 64-bit | LORegion Start Address (EL1)                     |
| LOREA_EL1              | 3   | 0   | C10 | C4  | 1   | See individual bit resets. | 64-bit | LORegion End Address (EL1)                       |
| LORN_EL1               | 3   | 0   | C10 | C4  | 2   | See individual bit resets. | 64-bit | LORegion Number (EL1)                            |
| LORC_EL1               | 3   | 0   | C10 | C4  | 3   | See individual bit resets. | 64-bit | LORegion Control (EL1)                           |
| LORID_EL1              | 3   | 0   | C10 | C4  | 7   | See individual bit resets. | 64-bit | LORegionID (EL1)                                 |
| VBAR_EL1               | 3   | 0   | C12 | C0  | 0   | See individual bit resets. | 64-bit | Vector Base Address Register (EL1)               |
| ISR_EL1                | 3   | 0   | C12 | C1  | 0   | See individual bit resets. | 64-bit | Interrupt Status Register                        |
| CONTEXTIDR_EL1         | 3   | 0   | C13 | C0  | 1   | See individual bit resets. | 64-bit | Context ID Register (EL1)                        |
| TPIDR_EL1              | 3   | 0   | C13 | C0  | 4   | See individual bit resets. | 64-bit | EL1 Software Thread ID Register                  |
| SCXTNUM_EL1            | 3   | 0   | C13 | C0  | 7   | See individual bit resets. | 64-bit | EL1 Read/Write Software Context Number           |
| IMP_CPUACTLR_EL1       | 3   | 0   | C15 | C1  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register (EL1)             |
| IMP_CPUACTLR2_EL1      | 3   | 0   | C15 | C1  | 1   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 2 (EL1)           |
| IMP_CPUACTLR3_EL1      | 3   | 0   | C15 | C1  | 2   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 3 (EL1)           |
| IMP_CPUACTLR4_EL1      | 3   | 0   | C15 | C1  | 3   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 4 (EL1)           |
| IMP_CPUECTLR_EL1       | 3   | 0   | C15 | C1  | 4   | See individual bit resets. | 64-bit | CPU Extended Control Register                    |
| IMP_CPUECTLR2_EL1      | 3   | 0   | C15 | C1  | 5   | See individual bit resets. | 64-bit | CPU Extended Control Register 2                  |
| IMP_CPUL2DIRTYLNCT_EL1 | 3   | 0   | C15 | C2  | 5   | See individual bit resets. | 64-bit | CPU L2 Dirty Line Count Register                 |
| IMP_CPUPWRCTLR_EL1     | 3   | 0   | C15 | C2  | 7   | See individual bit resets. | 64-bit | CPU Power Control Register                       |
| IMP_ATCR_EL1           | 3   | 0   | C15 | C7  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Translation Control Register (EL1) |
| IMP_CPUACTLR5_EL1      | 3   | 0   | C15 | C8  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 5 (EL1)           |
| IMP_CPUACTLR6_EL1      | 3   | 0   | C15 | C8  | 1   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 6 (EL1)           |
| IMP_CPUACTLR7_EL1      | 3   | 0   | C15 | C8  | 2   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 7 (EL1)           |
| IMP_CPUACTLR10_EL1     | 3   | 0   | C15 | C8  | 3   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 10 (EL1)          |



| Name               | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                           |
|--------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------|
| IMP_CPUACTLR8_EL1  | 3   | 0   | C15 | C8  | 5   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 8 (EL1)                |
| IMP_CPUACTLR9_EL1  | 3   | 0   | C15 | C8  | 6   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 9 (EL1)                |
| IMP_CPUACTLR11_EL1 | 3   | 0   | C15 | C8  | 7   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 11 (EL1)               |
| AIDR_EL1           | 3   | 1   | C0  | C0  | 7   | See individual bit resets. | 64-bit | Auxiliary ID Register                                 |
| NZCV               | 3   | 3   | C4  | C2  | 0   | See individual bit resets. | 64-bit | Condition Flags                                       |
| DAIF               | 3   | 3   | C4  | C2  | 1   | See individual bit resets. | 64-bit | Interrupt Mask Bits                                   |
| DIT                | 3   | 3   | C4  | C2  | 5   | See individual bit resets. | 64-bit | Data Independent Timing                               |
| SSBS               | 3   | 3   | C4  | C2  | 6   | See individual bit resets. | 64-bit | Speculative Store Bypass Safe                         |
| TCO                | 3   | 3   | C4  | C2  | 7   | See individual bit resets. | 64-bit | Tag Check Override                                    |
| FPCR               | 3   | 3   | C4  | C4  | 0   | See individual bit resets. | 64-bit | Floating-point Control Register                       |
| FPSR               | 3   | 3   | C4  | C4  | 1   | See individual bit resets. | 64-bit | Floating-point Status Register                        |
| TPIDR_ELO          | 3   | 3   | C13 | C0  | 2   | See individual bit resets. | 64-bit | ELO Read/Write Software Thread ID Register            |
| TPIDRRO_ELO        | 3   | 3   | C13 | C0  | 3   | See individual bit resets. | 64-bit | ELO Read-Only Software Thread ID Register             |
| SCXTNUM_ELO        | 3   | 3   | C13 | C0  | 7   | See individual bit resets. | 64-bit | ELO Read/Write Software Context Number                |
| ACTLR_EL2          | 3   | 4   | C1  | C0  | 1   | See individual bit resets. | 64-bit | Auxiliary Control Register (EL2)                      |
| HACR_EL2           | 3   | 4   | C1  | C1  | 7   | See individual bit resets. | 64-bit | Hypervisor Auxiliary Control Register                 |
| TTBR0_EL2          | 3   | 4   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Translation Table Base Register 0 (EL2)               |
| TTBR1_EL2          | 3   | 4   | C2  | C0  | 1   | See individual bit resets. | 64-bit | Translation Table Base Register 1 (EL2)               |
| TCR_EL2            | 3   | 4   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Translation Control Register (EL2)                    |
| VTTBR_EL2          | 3   | 4   | C2  | C1  | 0   | See individual bit resets. | 64-bit | Virtualization Translation Table Base Register        |
| VTCR_EL2           | 3   | 4   | C2  | C1  | 2   | See individual bit resets. | 64-bit | Virtualization Translation Control Register           |
| VSTTBR_EL2         | 3   | 4   | C2  | C6  | 0   | See individual bit resets. | 64-bit | Virtualization Secure Translation Table Base Register |
| VSTCR_EL2          | 3   | 4   | C2  | C6  | 2   | See individual bit resets. | 64-bit | Virtualization Secure Translation Control Register    |

| Name           | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                     |
|----------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------------------------|
| AFSR0_EL2      | 3   | 4   | C5  | C1  | 0   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 0 (EL2)                         |
| AFSR1_EL2      | 3   | 4   | C5  | C1  | 1   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 1 (EL2)                         |
| ESR_EL2        | 3   | 4   | C5  | C2  | 0   | See individual bit resets. | 64-bit | Exception Syndrome Register (EL2)                               |
| TFSR_EL2       | 3   | 4   | C5  | C6  | 0   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL2)                                 |
| FAR_EL2        | 3   | 4   | C6  | C0  | 0   | See individual bit resets. | 64-bit | Fault Address Register (EL2)                                    |
| HPFAR_EL2      | 3   | 4   | C6  | C0  | 4   | See individual bit resets. | 64-bit | Hypervisor IPA Fault Address Register                           |
| MAIR_EL2       | 3   | 4   | C10 | C2  | 0   | See individual bit resets. | 64-bit | Memory Attribute Indirection Register (EL2)                     |
| AMAIR_EL2      | 3   | 4   | C10 | C3  | 0   | See individual bit resets. | 64-bit | Auxiliary Memory Attribute Indirection Register (EL2)           |
| VBAR_EL2       | 3   | 4   | C12 | C0  | 0   | See individual bit resets. | 64-bit | Vector Base Address Register (EL2)                              |
| CONTEXTIDR_EL2 | 3   | 4   | C13 | C0  | 1   | See individual bit resets. | 64-bit | Context ID Register (EL2)                                       |
| TPIDR_EL2      | 3   | 4   | C13 | C0  | 2   | See individual bit resets. | 64-bit | EL2 Software Thread ID Register                                 |
| SCXTNUM_EL2    | 3   | 4   | C13 | C0  | 7   | See individual bit resets. | 64-bit | EL2 Read/Write Software Context Number                          |
| IMP_ATCR_EL2   | 3   | 4   | C15 | C7  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Translation Control Register (EL2)                |
| IMP_AVTCR_EL2  | 3   | 4   | C15 | C7  | 1   | See individual bit resets. | 64-bit | CPU Virtualization Auxiliary Translation Control Register (EL2) |
| ACTLR_EL3      | 3   | 6   | C1  | C0  | 1   | See individual bit resets. | 64-bit | Auxiliary Control Register (EL3)                                |
| SCR_EL3        | 3   | 6   | C1  | C1  | 0   | See individual bit resets. | 64-bit | Secure Configuration Register                                   |
| CPTR_EL3       | 3   | 6   | C1  | C1  | 2   | See individual bit resets. | 64-bit | Architectural Feature Trap Register (EL3)                       |
| MDCR_EL3       | 3   | 6   | C1  | C3  | 1   | See individual bit resets. | 64-bit | Monitor Debug Configuration Register (EL3)                      |
| TTBR0_EL3      | 3   | 6   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Translation Table Base Register 0 (EL3)                         |
| TCR_EL3        | 3   | 6   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Translation Control Register (EL3)                              |
| AFSR0_EL3      | 3   | 6   | C5  | C1  | 0   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 0 (EL3)                         |
| AFSR1_EL3      | 3   | 6   | C5  | C1  | 1   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 1 (EL3)                         |
| ESR_EL3        | 3   | 6   | C5  | C2  | 0   | See individual bit resets. | 64-bit | Exception Syndrome Register (EL3)                               |

| Name                    | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                             |
|-------------------------|-----|-----|-----|-----|-----|----------------------------|--------|---------------------------------------------------------|
| TFSR_EL3                | 3   | 6   | C5  | C6  | 0   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL3)                         |
| FAR_EL3                 | 3   | 6   | C6  | C0  | 0   | See individual bit resets. | 64-bit | Fault Address Register (EL3)                            |
| MAIR_EL3                | 3   | 6   | C10 | C2  | 0   | See individual bit resets. | 64-bit | Memory Attribute Indirection Register (EL3)             |
| AMAIR_EL3               | 3   | 6   | C10 | C3  | 0   | See individual bit resets. | 64-bit | Auxiliary Memory Attribute Indirection Register (EL3)   |
| VBAR_EL3                | 3   | 6   | C12 | C0  | 0   | See individual bit resets. | 64-bit | Vector Base Address Register (EL3)                      |
| RVBAR_EL3               | 3   | 6   | C12 | C0  | 1   | See individual bit resets. | 64-bit | Reset Vector Base Address Register (if EL3 implemented) |
| RMR_EL3                 | 3   | 6   | C12 | C0  | 2   | See individual bit resets. | 64-bit | Reset Management Register (EL3)                         |
| TPIDR_EL3               | 3   | 6   | C13 | C0  | 2   | See individual bit resets. | 64-bit | EL3 Software Thread ID Register                         |
| SCXTNUM_EL3             | 3   | 6   | C13 | C0  | 7   | See individual bit resets. | 64-bit | EL3 Read/Write Software Context Number                  |
| IMP_CPUL2SDIRTYLNCT_EL3 | 3   | 6   | C15 | C2  | 3   | See individual bit resets. | 64-bit | CPU L2 Secure Dirty Line Count Register                 |
| IMP_CPUACTLR_EL3        | 3   | 6   | C15 | C4  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register (EL3)                    |
| IMP_ATCR_EL3            | 3   | 6   | C15 | C7  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Translation Control Register (EL3)        |
| IMP_CPUPSELR_EL3        | 3   | 6   | C15 | C8  | 0   | See individual bit resets. | 64-bit | Selected Instruction Private Select Register            |
| IMP_CPUPCR_EL3          | 3   | 6   | C15 | C8  | 1   | See individual bit resets. | 64-bit | Selected Instruction Private Control Register           |
| IMP_CPUPOR_EL3          | 3   | 6   | C15 | C8  | 2   | See individual bit resets. | 64-bit | Selected Instruction Private Opcode Register            |
| IMP_CPUPMR_EL3          | 3   | 6   | C15 | C8  | 3   | See individual bit resets. | 64-bit | Selected Instruction Private Mask Register              |
| IMP_CPUPOR2_EL3         | 3   | 6   | C15 | C8  | 4   | See individual bit resets. | 64-bit | Selected Instruction Private Opcode Register 2          |
| IMP_CPUPMR2_EL3         | 3   | 6   | C15 | C8  | 5   | See individual bit resets. | 64-bit | Selected Instruction Private Mask Register 2            |
| IMP_CPUPFR_EL3          | 3   | 6   | C15 | C8  | 6   | See individual bit resets. | 64-bit | Selected Instruction Private Flag Register              |

## 16. Debug

The DSU-120 DynamiQ™ cluster provides a debug system that supports both self-hosted and external debug. It has an external DebugBlock component and integrates various CoreSight debug related components.

The CoreSight debug related components are split into two groups, with some components in the DSU-120 DynamiQ™ cluster, and others in the separate DebugBlock.

The DebugBlock is a dedicated debug component in the DSU-120, separate from the cluster. The DebugBlock operates within a separate power domain, which enables connection to a debugger to be maintained when the cores and the DSU-120 DynamiQ™ cluster are both powered down.

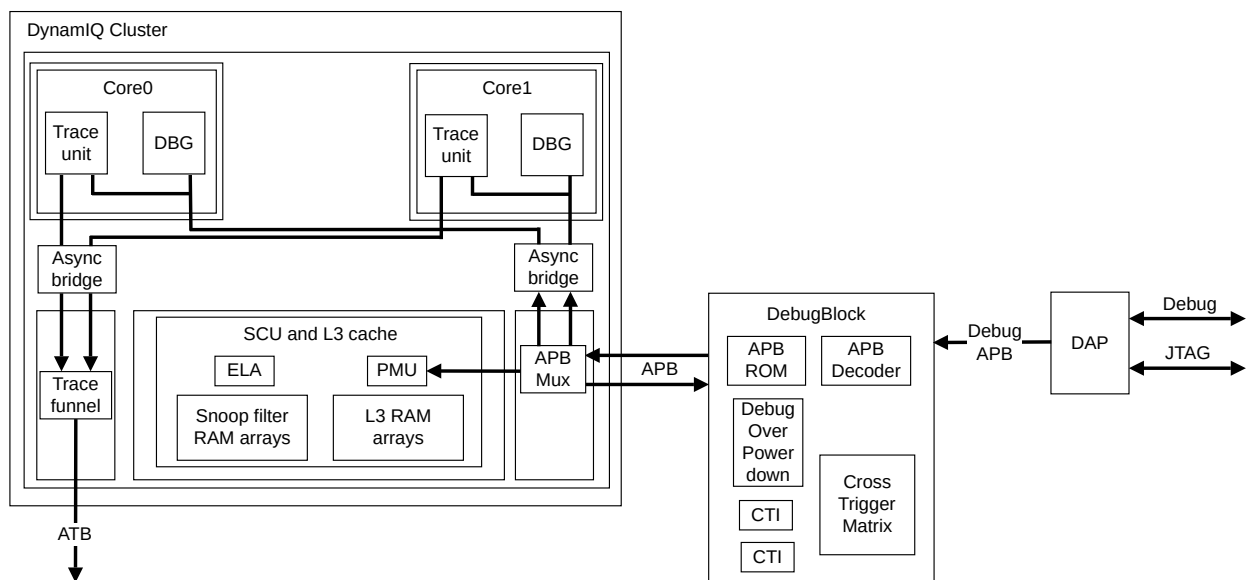
The connection between the cluster and the DebugBlock consists of a pair of *Advanced Peripheral Bus* (APB) interfaces, one in each direction. All debug traffic, except the authentication interface, takes place over this interface as read or write APB transactions. This debug traffic includes register reads, register writes, and *Cross Trigger Interface* (CTI) triggers.

The debug system implements the following CoreSight debug components:

- Per-core trace unit, integrated into the CoreSight subsystem
- Per-core CTI, contained in the DebugBlock
- *Cross Trigger Matrix* (CTM)
- Debug control provided by AMBA® APB interface to the DebugBlock

The following figure shows how the debug system is implemented with the DSU-120 DynamiQ™ cluster.

**Figure 16-1: DSU-120 DynamiQ™ cluster debug components**



The primary debug APB interface on the DebugBlock controls the debug components. The APB decoder decodes the requests on this bus before they are sent to the appropriate component in the DebugBlock or in the DSU-120 DynamIQ™ cluster. The per-core CTIs are connected to a CTM.

Each core contains a debug component that the debug APB bus accesses. The cores support debug over powerdown using modules in the DebugBlock that mirror key core information. These modules allow access to debug over powerdown CoreSight™ registers while the core is powered down.

The trace unit in each core outputs trace, which is funneled in the DSU-120 DynamIQ™ cluster down to a single AMBA® 4 ATBv1.1 interface.

See *Debug* in the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#) for more information about the DSU-120 DynamIQ™ cluster debug components.

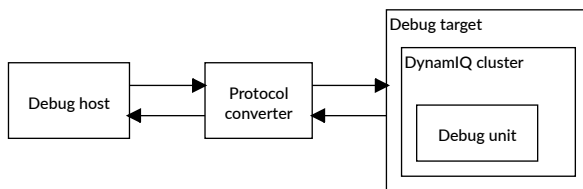
The Cortex®-X925 core also supports direct access to internal memory, that is, cache debug. Direct access to internal memory allows software to read the internal memory that the L1 and L2 cache and *Translation Lookaside Buffer* (TLB) structures use. See [9. Direct access to internal memory](#) on page 70 for more information.

## 16.1 Supported debug methods

The DSU-120 DynamIQ™ cluster along with its associated cores is part of a debug system that supports both self-hosted and external debug.

The following figure shows a typical external debug system.

**Figure 16-2: External debug system**



### Debug host

A computer, for example a personal computer, that is running a software debugger such as the Arm® Debugger. You can use the debug host to issue high-level commands. For example, you can set a breakpoint at a certain location or examine the contents of a memory address.

### Protocol converter

The debug host sends messages to the debug target using an interface such as Ethernet. However, the debug target typically implements a different interface protocol. A device such as DSTREAM is required to convert between the two protocols.

## Debug target

The lowest level of the system implements system support for the protocol converter to access the debug unit. For DSU-120 based devices, the mechanism used to access the debug unit is based on the CoreSight architecture. The DSU-120 DebugBlock is accessed using an *Advanced Peripheral Bus* (APB) interface and the debug accesses are then directed to the selected Cortex®-X925 core inside the DSU-120 DynamiQ™ cluster. An example of a debug target is a development system with a test chip or a silicon part with a Cortex®-X925 core.

## Debug unit

Helps debugging software that is running on the core:

- DSU-120 and external hardware based around the core
- Operating systems
- Application software

With the debug unit, you can:

- Stop program execution.
- Examine and alter process and coprocessor state.
- Examine and alter memory and the state of the input or output peripherals.
- Restart the *Processing Element* (PE).

For self-hosted debug, the debug target runs debug monitor software that runs on the core in the DSU-120 DynamiQ™ cluster. This way, it does not require expensive interface hardware to connect a second host computer.

## 16.2 Debug register interfaces

The Cortex®-X925 core implements the Arm®v9.2-A Debug architecture. It also supports the Arm®v8.4-A Debug architecture and Arm®v8.3-A Debug over powerdown.

The Debug architecture defines a set of Debug registers. The Debug register interfaces provide access to these registers either from software running on the core or from an external debugger. See *Debug* in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#) for more information.

### Related information

[4.7 Debug over powerdown](#) on page 49

### 16.2.1 Core interfaces

System register access allows the Cortex®-X925 core to access certain Debug registers directly. The Debug register interfaces provide access to these registers either from software running on the core or from an external debugger.

Access to the Debug registers is partitioned as follows:

## Debug

This function is both system register based and memory-mapped. You can access the Debug register map using the *Advanced Peripheral Bus* (APB) slave port that connects into the DebugBlock of the *DynamlQ™ Shared Unit-120*.

## Performance monitoring

This function is system register based and memory-mapped. You can access the performance monitor registers using the APB slave port that connects into the DebugBlock of the *DynamlQ™ Shared Unit-120*.

## Trace

This function is system register based and memory-mapped. You can access the trace unit registers using the APB slave port that connects into the DebugBlock of the *DynamlQ™ Shared Unit-120*.

## Statistical profiling

This function is system register based.

## ELA registers

You can access the *Embedded Logic Analyzer* (ELA) registers using the APB slave port that connects into the DebugBlock of the *DynamlQ™ Shared Unit-120*.

The ELA-600 is licensed separately.



This function is memory-mapped and is not accessible using System registers.

---

For information on APB slave port interface, see the *Debug* chapter or the *Interfaces* section in the *Technical overview* chapter of the [Arm® DynamlQ™ Shared Unit-120 Technical Reference Manual](#).

## 16.2.2 Effects of resets on debug registers

The complexporeset\_n and complexreset\_n signals of the core affect the debug registers.

complexporeset\_n maps to a Cold reset that covers reset of the core logic and the integrated debug functionality. This signal initializes the core logic, including the trace unit, breakpoint, watchpoint logic, performance monitor, and debug logic.

complexreset\_n maps to a Warm reset that covers reset of the core logic. This signal resets some of the debug and performance monitor logic.

### 16.2.3 Breakpoints and watchpoints

The Cortex®-X925 core supports six breakpoints, four watchpoints, and a standard *Debug Communications Channel* (DCC).

A breakpoint consists of a breakpoint control register and a breakpoint value register. These two registers are referred to as a *Breakpoint Register Pair* (BRP). Four of the breakpoints (BRP 0-3) match only to the *Virtual Address* (VA) and the other two (BRP 4 and 5) match against either the VA or context ID, or the *Virtual Machine Identifier* (VMID).

You can use watchpoints to stop your target when a specific memory address is accessed by your program. All the watchpoints can be linked to two breakpoints (BRP 4 and 5) to enable a memory request to be trapped in a given process context.

## 16.3 Debug events

A debug event can be either a software debug event or a Halting debug event.

The Cortex®-X925 core responds to a debug event in one of the following ways:

- It ignores the debug event
- It takes a debug exception
- It enters debug state

In the Cortex®-X925 core, watchpoint debug events are always synchronous. Memory hint instructions and cache clean operations, except `DC ZVA`, and `DC IVA`, do not generate watchpoint debug events. Store exclusive instructions generate a watchpoint debug event even when the check for the control of exclusive monitor fails. Atomic `CAS` instructions generate a watchpoint debug event even when the compare operation fails.

A Cold reset sets the Debug OS Lock. For the debug events and debug register accesses to operate normally, the Debug OS Lock must be cleared.

## 16.4 Debug memory map and debug signals

The debug memory map and debug signals are handled at the DSU-120 DynamiQ™ cluster level.

See *Debug* and *ROM tables* in the [Arm® DynamiQ™ Shared Unit-120 Technical Reference Manual](#).



## 16.5 ROM table

The Cortex®-X925 core includes a ROM table that contains a list of components in the system. Debuggers must use the ROM table to determine which CoreSight components are implemented.

The ROM table is a CoreSight debug related component that aids system debug along with CoreSight SoC and is for the Cortex®-X925 core. There is one ROM table for each core and ROM tables comply with the [Arm® CoreSight™ Architecture Specification v3.0](#).

The *DynamlQ™ Shared Unit-120* has its own ROM tables, one for the cluster and one for the DebugBlock, and has entry points in the cluster ROM table for the ROM tables belonging to each core. See *ROM tables* in the [Arm® DynamlQ™ Shared Unit-120 Technical Reference Manual](#) for more information.

The Cortex®-X925 core ROM table includes the following entries:

**Table 16-1: Core ROM table**

| Offset | Name      | Description     |
|--------|-----------|-----------------|
| 0x0000 | ROMENTRY0 | Core debug      |
| 0x0004 | ROMENTRY1 | Core PMU        |
| 0x0008 | ROMENTRY2 | Core trace unit |
| 0x000C | ROMENTRY3 | Optional ELA    |

### Related information

[16.10 External CoreROM registers](#) on page 117

## 16.6 CoreSight component identification

Each component associated with the Cortex®-X925 core has a unique set of CoreSight™ ID values. The following table shows these values.

**Table 16-2: Cortex®-X925 core CoreSight™ component identification**

| Component  | Peripheral ID | Component ID | DevType | DevArch    | Core revision |
|------------|---------------|--------------|---------|------------|---------------|
| DBG        | 0x04200BBD85  | 0xB105900D   | 0x15    | 0x47709A15 | r0p2          |
| PMU        |               |              | 0x16    | 0x47702A16 |               |
| Trace unit |               |              | 0x13    | 0x47715A13 |               |
| ROM table  |               |              | 0x00    | 0x47700AF7 |               |

## 16.7 CTI register identification values

The Cortex®-X925 core *Cross Trigger Interface* (CTI) registers are located in the DebugBlock of the DSU-120.

For the cluster and core CTI register names and descriptions, see *External CTI registers* in the *Debug* chapter of the *Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual*. Only the core CTI register peripheral ID values will differ from the cluster CTI register peripheral ID values.

The core CTI register peripheral ID values are listed in the following table.

**Table 16-3: Core CTI register peripheral ID values**

| Register | Bitfield position | Bitfield name | Value  |
|----------|-------------------|---------------|--------|
| CTIPIDR1 | [7:4]             | DES_0         | 0b1011 |
|          | [3:0]             | PART_1        | 0b1101 |
| CTIPIDRO | [7:0]             | PART_0        | 0x85   |

## 16.8 AArch64 Debug registers

The following summary table provides an overview of all Debug registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the *Arm® Architecture Reference Manual for A-profile architecture*.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the *Arm® Architecture Reference Manual for A-profile architecture*.

**Table 16-4: Debug registers summary**

| Name        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                             |
|-------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------|
| OSDTRRX_EL1 | 2   | 0   | C0  | C0  | 2   | See individual bit resets. | 64-bit | OS Lock Data Transfer Register, Receive |
| DBGBVR0_EL1 | 2   | 0   | C0  | C0  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers        |
| DBGBCR0_EL1 | 2   | 0   | C0  | C0  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers      |
| DBGWVR0_EL1 | 2   | 0   | C0  | C0  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers        |
| DBGWCR0_EL1 | 2   | 0   | C0  | C0  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers      |
| DBGBVR1_EL1 | 2   | 0   | C0  | C1  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers        |
| DBGBCR1_EL1 | 2   | 0   | C0  | C1  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers      |
| DBGWVR1_EL1 | 2   | 0   | C0  | C1  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers        |
| DBGWCR1_EL1 | 2   | 0   | C0  | C1  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers      |
| MDCCINT_EL1 | 2   | 0   | C0  | C2  | 0   | See individual bit resets. | 64-bit | Monitor DCC Interrupt Enable Register   |
| MDSCR_EL1   | 2   | 0   | C0  | C2  | 2   | See individual bit resets. | 64-bit | Monitor Debug System Control Register   |

| Name              | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|-------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| DBGBVR2_EL1       | 2   | 0   | C0  | C2  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR2_EL1       | 2   | 0   | C0  | C2  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| DBGWVR2_EL1       | 2   | 0   | C0  | C2  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers           |
| DBGWCR2_EL1       | 2   | 0   | C0  | C2  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers         |
| OSDTRTX_EL1       | 2   | 0   | C0  | C3  | 2   | See individual bit resets. | 64-bit | OS Lock Data Transfer Register, Transmit   |
| DBGBVR3_EL1       | 2   | 0   | C0  | C3  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR3_EL1       | 2   | 0   | C0  | C3  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| DBGWVR3_EL1       | 2   | 0   | C0  | C3  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers           |
| DBGWCR3_EL1       | 2   | 0   | C0  | C3  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers         |
| DBGBVR4_EL1       | 2   | 0   | C0  | C4  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR4_EL1       | 2   | 0   | C0  | C4  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| DBGBVR5_EL1       | 2   | 0   | C0  | C5  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR5_EL1       | 2   | 0   | C0  | C5  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| OSECCR_EL1        | 2   | 0   | C0  | C6  | 2   | See individual bit resets. | 64-bit | OS Lock Exception Catch Control Register   |
| MDRAR_EL1         | 2   | 0   | C1  | C0  | 0   | See individual bit resets. | 64-bit | Monitor Debug ROM Address Register         |
| OSLAR_EL1         | 2   | 0   | C1  | C0  | 4   | See individual bit resets. | 64-bit | OS Lock Access Register                    |
| OSLSR_EL1         | 2   | 0   | C1  | C1  | 4   | See individual bit resets. | 64-bit | OS Lock Status Register                    |
| OSDLR_EL1         | 2   | 0   | C1  | C3  | 4   | See individual bit resets. | 64-bit | OS Double Lock Register                    |
| DBGPRCR_EL1       | 2   | 0   | C1  | C4  | 4   | See individual bit resets. | 64-bit | Debug Power Control Register               |
| DBGCLAIMSET_EL1   | 2   | 0   | C7  | C8  | 6   | See individual bit resets. | 64-bit | Debug CLAIM Tag Set Register               |
| DBGCLAIMCLR_EL1   | 2   | 0   | C7  | C9  | 6   | See individual bit resets. | 64-bit | Debug CLAIM Tag Clear Register             |
| DBGAUTHSTATUS_EL1 | 2   | 0   | C7  | C14 | 6   | See individual bit resets. | 64-bit | Debug Authentication Status Register       |
| MDCCSR_ELO        | 2   | 3   | C0  | C1  | 0   | See individual bit resets. | 64-bit | Monitor DCC Status Register                |
| DBGDTR_ELO        | 2   | 3   | C0  | C4  | 0   | See individual bit resets. | 64-bit | Debug Data Transfer Register, half-duplex  |
| DBGDTRRX_ELO      | 2   | 3   | C0  | C5  | 0   | See individual bit resets. | 64-bit | Debug Data Transfer Register, Receive      |
| DBGDTRTX_ELO      | 2   | 3   | C0  | C5  | 0   | See individual bit resets. | 64-bit | Debug Data Transfer Register, Transmit     |
| TRFCR_EL1         | 3   | 0   | C1  | C2  | 1   | See individual bit resets. | 64-bit | Trace Filter Control Register (EL1)        |
| MDCR_EL2          | 3   | 4   | C1  | C1  | 1   | See individual bit resets. | 64-bit | Monitor Debug Configuration Register (EL2) |
| TRFCR_EL2         | 3   | 4   | C1  | C2  | 1   | See individual bit resets. | 64-bit | Trace Filter Control Register (EL2)        |
| IMP_IDATA0_EL3    | 3   | 6   | C15 | C0  | 0   | See individual bit resets. | 64-bit | Instruction Register 0                     |
| IMP_IDATA1_EL3    | 3   | 6   | C15 | C0  | 1   | See individual bit resets. | 64-bit | Instruction Register 1                     |
| IMP_IDATA2_EL3    | 3   | 6   | C15 | C0  | 2   | See individual bit resets. | 64-bit | Instruction Register 2                     |
| IMP_DDATA0_EL3    | 3   | 6   | C15 | C1  | 0   | See individual bit resets. | 64-bit | Data Register 0                            |
| IMP_DDATA1_EL3    | 3   | 6   | C15 | C1  | 1   | See individual bit resets. | 64-bit | Data Register 1                            |
| IMP_DDATA2_EL3    | 3   | 6   | C15 | C1  | 2   | See individual bit resets. | 64-bit | Data Register 2                            |

## 16.9 External Debug registers

The following summary table provides an overview of all memory-mapped Debug registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 16-5: Debug registers summary**

| Offset | Name                   | Reset                      | Width  | Description                                     |
|--------|------------------------|----------------------------|--------|-------------------------------------------------|
| 0x020  | EDESR                  | See individual bit resets. | 32-bit | External Debug Event Status Register            |
| 0x024  | EDECR                  | See individual bit resets. | 32-bit | External Debug Execution Control Register       |
| 0x030  | EDWAR                  | See individual bit resets. | 64-bit | External Debug Watchpoint Address Register      |
| 0x080  | DBGDTRRX_ELO           | See individual bit resets. | 32-bit | Debug Data Transfer Register, Receive           |
| 0x084  | EDITR                  | See individual bit resets. | 32-bit | External Debug Instruction Transfer Register    |
| 0x088  | EDSCR                  | See individual bit resets. | 32-bit | External Debug Status and Control Register      |
| 0x08C  | DBGDTRTX_ELO           | See individual bit resets. | 32-bit | Debug Data Transfer Register, Transmit          |
| 0x090  | <a href="#">EDRCR</a>  | See individual bit resets. | 32-bit | External Debug Reserve Control Register         |
| 0x098  | EDECCR                 | See individual bit resets. | 32-bit | External Debug Exception Catch Control Register |
| 0x300  | OSLAR_EL1              | See individual bit resets. | 32-bit | OS Lock Access Register                         |
| 0x310  | <a href="#">EDPRCR</a> | See individual bit resets. | 32-bit | External Debug Power/Reset Control Register     |
| 0x314  | EDPRSR                 | See individual bit resets. | 32-bit | External Debug Processor Status Register        |
| 0x400  | DBGBVR0_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                |
| 0x408  | DBGBCR0_EL1            | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers              |
| 0x410  | DBGBVR1_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                |
| 0x418  | DBGBCR1_EL1            | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers              |
| 0x420  | DBGBVR2_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                |
| 0x428  | DBGBCR2_EL1            | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers              |
| 0x430  | DBGBVR3_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                |
| 0x438  | DBGBCR3_EL1            | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers              |
| 0x440  | DBGBVR4_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                |
| 0x448  | DBGBCR4_EL1            | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers              |
| 0x450  | DBGBVR5_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                |
| 0x458  | DBGBCR5_EL1            | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers              |
| 0x800  | DBGWVR0_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                |
| 0x808  | DBGWCR0_EL1            | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers              |
| 0x810  | DBGWVR1_EL1 [63:0]     | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                |

| Offset | Name                      | Reset                      | Width  | Description                                         |
|--------|---------------------------|----------------------------|--------|-----------------------------------------------------|
| 0x818  | DBGWCR1_EL1               | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0x820  | DBGWVR2_EL1 [63:0]        | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                    |
| 0x828  | DBGWCR2_EL1               | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0x830  | DBGWVR3_EL1 [63:0]        | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                    |
| 0x838  | DBGWCR3_EL1               | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0xD00  | <a href="#">MIDR_EL1</a>  | See individual bit resets. | 32-bit | Main ID Register                                    |
| 0xD20  | <a href="#">EDPFR</a>     | See individual bit resets. | 64-bit | External Debug Processor Feature Register           |
| 0xD28  | <a href="#">EDDFR</a>     | See individual bit resets. | 64-bit | External Debug Feature Register                     |
| 0xD48  | <a href="#">EDDFR1</a>    | See individual bit resets. | 64-bit | External Debug Feature Register 1                   |
| 0xD60  | EDAA32PFR                 | See individual bit resets. | 64-bit | External Debug Auxiliary Processor Feature Register |
| 0xF00  | EDITCTRL                  | See individual bit resets. | 32-bit | External Debug Integration mode Control register    |
| 0xFA0  | DBGCLAIMSET_EL1           | See individual bit resets. | 32-bit | Debug CLAIM Tag Set Register                        |
| 0xFA4  | DBGCLAIMCLR_EL1           | See individual bit resets. | 32-bit | Debug CLAIM Tag Clear Register                      |
| 0xFA8  | EDDEVAFF0                 | See individual bit resets. | 32-bit | External Debug Device Affinity register 0           |
| 0xFAC  | EDDEVAFF1                 | See individual bit resets. | 32-bit | External Debug Device Affinity register 1           |
| 0xFB0  | EDLAR                     | See individual bit resets. | 32-bit | External Debug Lock Access Register                 |
| 0xFB4  | EDLSR                     | See individual bit resets. | 32-bit | External Debug Lock Status Register                 |
| 0xFB8  | DBGAUTHSTATUS_EL1         | See individual bit resets. | 32-bit | Debug Authentication Status Register                |
| 0xFBC  | <a href="#">EDDEVARCH</a> | See individual bit resets. | 32-bit | External Debug Device Architecture Register         |
| 0xFC0  | <a href="#">EDDEVID2</a>  | See individual bit resets. | 32-bit | External Debug Device ID register 2                 |
| 0xFC4  | <a href="#">EDDEVID1</a>  | See individual bit resets. | 32-bit | External Debug Device ID Register 1                 |
| 0xFC8  | <a href="#">EDDEVID</a>   | See individual bit resets. | 32-bit | External Debug Device ID register 0                 |
| 0xFCC  | <a href="#">EDDEVTYPE</a> | See individual bit resets. | 32-bit | External Debug Device Type register                 |
| 0xFD0  | <a href="#">EDPIDR4</a>   | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 4 |
| 0xFE0  | <a href="#">EDPIDR0</a>   | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 0 |
| 0xFE4  | <a href="#">EDPIDR1</a>   | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 1 |
| 0xFE8  | <a href="#">EDPIDR2</a>   | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 2 |
| 0xFEC  | <a href="#">EDPIDR3</a>   | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 3 |
| 0xFF0  | <a href="#">EDCIDR0</a>   | See individual bit resets. | 32-bit | External Debug Component Identification Register 0  |
| 0xFF4  | <a href="#">EDCIDR1</a>   | See individual bit resets. | 32-bit | External Debug Component Identification Register 1  |
| 0xFF8  | <a href="#">EDCIDR2</a>   | See individual bit resets. | 32-bit | External Debug Component Identification Register 2  |
| 0xFFC  | <a href="#">EDCIDR3</a>   | See individual bit resets. | 32-bit | External Debug Component Identification Register 3  |

## 16.10 External CoreROM registers

The following summary table provides an overview of all memory-mapped CoreROM registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 16-6: CoreROM registers summary**

| Offset | Name                               | Reset                      | Width  | Description                                         |
|--------|------------------------------------|----------------------------|--------|-----------------------------------------------------|
| 0x000  | <a href="#">COREROM_ROMENTRY0</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 0                              |
| 0x004  | <a href="#">COREROM_ROMENTRY1</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 1                              |
| 0x008  | <a href="#">COREROM_ROMENTRY2</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 2                              |
| 0x00C  | <a href="#">COREROM_ROMENTRY3</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 3                              |
| 0xFB8  | <a href="#">COREROM_AUTHSTATUS</a> | See individual bit resets. | 32-bit | Core ROM table Authentication Status Register       |
| 0xFBC  | <a href="#">COREROM_DEVARCH</a>    | See individual bit resets. | 32-bit | Core ROM table Device Architecture Register         |
| 0xFCC  | <a href="#">COREROM_DEVTYPE</a>    | See individual bit resets. | 32-bit | Core ROM table Device Type Register                 |
| 0xFD0  | <a href="#">COREROM_PIDR4</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 4 |
| 0xFE0  | <a href="#">COREROM_PIDR0</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 0 |
| 0xFE4  | <a href="#">COREROM_PIDR1</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 1 |
| 0xFE8  | <a href="#">COREROM_PIDR2</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 2 |
| 0xFEC  | <a href="#">COREROM_PIDR3</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 3 |
| 0xFF0  | <a href="#">COREROM_CIDR0</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 0  |
| 0xFF4  | <a href="#">COREROM_CIDR1</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 1  |
| 0xFF8  | <a href="#">COREROM_CIDR2</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 2  |
| 0xFFC  | <a href="#">COREROM_CIDR3</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 3  |

# 17. Performance Monitors Extension support

The Cortex®-X925 core implements the Performance Monitors Extension, including Arm®v8.4-A, Arm®v8.5-A and Arm®v8.7-A performance monitoring features.

The Cortex®-X925 core *Performance Monitoring Unit* (PMU):

- Collects events through an event interface from other units in the design. These events are used as triggers for event counters.
- Supports cycle counters through the Performance Monitors Control Register.
- Implements PMU snapshots for context samples.
- Provides six or 31 PMU 64-bit counters that count any of the events available in the core. The absolute counts that are recorded might vary because of pipeline effects. This variation has negligible effect except in cases where the counters are enabled for a very short time.

You can program the PMU using either the System registers or the external Debug APB interface.

## 17.1 Performance monitors events

The Cortex®-X925 core *Performance Monitoring Unit* (PMU) collects events from other units in the design and uses numbers to reference these events.

### Common Event PMU events

The following table shows the Cortex®-X925 core performance monitors events that are generated and the numbers that the PMU uses to reference the events. The table also shows the bit position of each event on the event bus. Event numbers that are not listed are reserved.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about these PMU events.



Note

Unless otherwise indicated, each of these events can be exported to the trace unit and selected in accordance with the *Arm® Embedded Trace Extension*.

**Table 17-1: Common Event PMU events**

| Event number | Mnemonic | Description                                                                                                                                                                        |
|--------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0000       | SW_INCR  | Instruction architecturally executed, Condition code check pass, software increment<br><br>This event counts any instruction architecturally executed (condition code check pass). |

| Event number | Mnemonic         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0001       | L1I_CACHE_REFILL | <p>Level 1 instruction cache refill</p> <p>This event counts any instruction fetch which misses in the cache.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>• Cache maintenance instructions</li> <li>• Non-cacheable accesses</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                         |
| 0x0002       | L1I_TLB_REFILL   | <p>Level 1 instruction TLB refill</p> <p>This event counts any refill of the L1 instruction TLB from the MMU Translation Cache (MMUTC). This includes refills that result in a translation fault.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>• TLB maintenance instructions</li> </ul> <p>This event counts regardless of whether the MMU is enabled.</p>                                                                                                                                                                                                                                                                      |
| 0x0003       | L1D_CACHE_REFILL | <p>Level 1 data cache refill</p> <p>This event counts any load or store operation or translation table walk that causes data to be read from outside the L1 cache, including accesses which do not allocate into L1.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>• Cache maintenance instructions and prefetches.</li> <li>• Stores of an entire cache line, even if they make a coherency request outside the L1.</li> <li>• Partial cache line writes which do not allocate into the L1 cache.</li> <li>• Non-cacheable accesses.</li> </ul> <p>This event counts the sum of L1D_CACHE_REFILL_RD and L1D_CACHE_REFILL_WR.</p> |
| 0x0004       | L1D_CACHE        | <p>Level 1 data cache access</p> <p>This event counts any load or store operation or translation table walk that looks up in the L1 data cache. In particular, any access that could count the L1D_CACHE_REFILL event causes this event to count.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>• Cache maintenance instructions and prefetches.</li> <li>• Non-cacheable accesses.</li> </ul> <p>This event counts the sum of L1D_CACHE_RD and L1D_CACHE_WR.</p>                                                                                                                                                                 |
| 0x0005       | L1D_TLB_REFILL   | <p>Level 1 data TLB refill</p> <p>This event counts any refill of the data L1 TLB from the L2 TLB. This includes refills which result in a translation fault.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>• TLB maintenance instructions.</li> </ul> <p>This event counts regardless of whether the MMU is enabled.</p>                                                                                                                                                                                                                                                                                                         |



| Event number | Mnemonic          | Description                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0008       | INST_RETIRED      | Instruction architecturally executed<br><br>This event counts all retired instructions, including ones that fail their condition check.                                                                                                                                                                                                                                               |
| 0x0009       | EXC_TAKEN         | Exception taken<br><br>The counter counts each exception taken.                                                                                                                                                                                                                                                                                                                       |
| 0x000A       | EXC_RETURN        | Instruction architecturally executed, Condition code check pass, exception return                                                                                                                                                                                                                                                                                                     |
| 0x000B       | CID_WRITE_RETIRED | Instruction architecturally executed, Condition code check pass, write to CONTEXTIDR<br><br>This event only counts writes using the CONTEXTIDR_EL1 mnemonic.<br><br>Writes to CONTEXTIDR_EL12 and CONTEXTIDR_EL2 are not counted.                                                                                                                                                     |
| 0x000D       | BR_IMMED_RETIRED  | Instruction architecturally executed, immediate branch<br><br>This event counts all branches decoded as immediate branches, taken or not, and popped from the branch monitor. This excludes exception entries, debug entries, and CCFAIL branches.                                                                                                                                    |
| 0x000E       | BR_RETURN_RETIRED | Instruction architecturally executed, Condition code check pass, procedure return                                                                                                                                                                                                                                                                                                     |
| 0x0010       | BR_MIS_PRED       | Mispredicted or not predicted branch speculatively executed<br><br>This event counts any predictable branch instruction which is mispredicted either due to dynamic misprediction or because the MMU is off and the branches are statically predicted not taken.                                                                                                                      |
| 0x0011       | CPU_CYCLES        | Cycle                                                                                                                                                                                                                                                                                                                                                                                 |
| 0x0012       | BR_PRED           | Predictable branch speculatively executed<br><br>This event counts all predictable branches.                                                                                                                                                                                                                                                                                          |
| 0x0013       | MEM_ACCESS        | Data memory access<br><br>This event counts memory accesses due to load or store instructions.<br><br>The following instructions are not counted: <ul style="list-style-type: none"> <li>• Instruction fetches.</li> <li>• Cache maintenance instructions.</li> <li>• Translation table walks or prefetches.</li> </ul> This event counts the sum of MEM_ACCESS_RD and MEM_ACCESS_WR. |
| 0x0014       | L1I_CACHE         | Level 1 instruction cache access<br><br>This event counts any instruction fetch which accesses the L1 instruction cache.<br><br>The following instructions are not counted: <ul style="list-style-type: none"> <li>• Cache maintenance instructions.</li> <li>• Non-cacheable accesses.</li> </ul>                                                                                    |

| Event number | Mnemonic           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0015       | L1D_CACHE_WB       | <p>Level 1 data cache write-back</p> <p>This event counts any write-back of data from the L1 data cache to L2 or L3. The event counts both victim line evictions and snoops, including cache maintenance operations.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>Invalidations which do not result in data being transferred out of the L1.</li> <li>Full-line writes which write to L2 without writing L1, such as write-streaming mode.</li> </ul>                                                                                     |
| 0x0016       | L2D_CACHE          | <p>Level 2 data cache access</p> <ul style="list-style-type: none"> <li>If the core is configured with a per-core L2 cache, this event counts any transaction from L1 which looks up in the L2 cache, and any writeback from the L1 to the L2. Snoops from outside the core and cache maintenance operations are not counted.</li> <li>If the core is not configured with a per-core L2 cache, this event counts the cluster cache event, as defined by L3D_CACHE.</li> <li>If neither a per-core cache nor a cluster cache are configured, then this event is not implemented.</li> </ul> |
| 0x0017       | L2D_CACHE_REFILL   | <p>Level 2 data cache refill</p> <ul style="list-style-type: none"> <li>If the core is configured with a per-core L2 cache, this event counts any Cacheable transaction from L1 which causes data to be read from outside the core. L2 refills caused by stashes into L2 are not meant to be counted.</li> <li>If the core is not configured with a per-core L2 cache, this event counts the cluster cache event, as defined by L3D_CACHE_REFILL.</li> <li>If neither a per-core cache nor a cluster cache are configured, then this event is not implemented.</li> </ul>                  |
| 0x0018       | L2D_CACHE_WB       | <p>Level 2 data cache write-back</p> <p>If the core is configured with a per-core L2 cache, this event counts any write-back of data from the L2 cache to a location outside the core. The event includes snoops to the L2 which return data, regardless of whether they cause an invalidation. Invalidations from the L2 which do not write data outside of the core and snoops which return data from the L1 are not counted.</p> <ul style="list-style-type: none"> <li>If the core is not configured with a per-core L2 cache, this event is not implemented.</li> </ul>               |
| 0x0019       | BUS_ACCESS         | <p>This event counts for every beat of data that is transferred over the data channels between the core and the SCU. If both read and write data beats are transferred on a given cycle, this event is counted twice on that cycle.</p> <p>This event counts the sum of BUS_ACCESS_RD and BUS_ACCESS_WR.</p>                                                                                                                                                                                                                                                                               |
| 0x001A       | MEMORY_ERROR       | Local memory error                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 0x001B       | INST_SPEC          | <p>Operation speculatively executed</p> <p>Counts operations that have been speculatively executed.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 0x001C       | TTBR_WRITE_RETIRED | <p>Instruction architecturally executed, condition code check pass, write to TTBR</p> <p>This event only counts writes to TTBR0/TTBR1 in AArch32 and TTBR0_EL1/TTBR1_EL1 in AArch64.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>Accesses to TTBR0_EL12/TTBR1_EL12 or TTBR0_EL2/TTBR1_EL2.</li> </ul>                                                                                                                                                                                                                                    |

| Event number | Mnemonic            | Description                                                                                                                                                                                                                                                                                                                              |
|--------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x1D         | BUS_CYCLES          | Bus cycle                                                                                                                                                                                                                                                                                                                                |
| 0x001E       | CHAIN               | For odd-numbered counters, this event increments the count by one for each overflow of the preceding even-numbered counter. For even numbered counters, there is no increment.                                                                                                                                                           |
| 0x0020       | L2D_CACHE_ALLOCATE  | Level 2 data cache allocation without refill<br><br>This event counts any full cache line write into the L2 cache which does not cause a linefill, including write-backs from L1 to L2 and full-line writes which do not allocate into L1.                                                                                               |
| 0x0021       | BR_RETIRED          | Instruction architecturally executed, branch<br><br>This event counts all branches, taken or not, popped from the branch monitor. This excludes exception entries, debug entries, and CCFail branches. In the core, an ISB is a branch, and even micro architectural ISBs are counted.                                                   |
| 0x0022       | BR_MIS_PRED_RETIRED | Instruction architecturally executed, mispredicted branch<br><br>This event counts any branch that is counted by BR_RETIRED which is not correctly predicted and causes a pipeline clean.                                                                                                                                                |
| 0x0023       | STALL_FRONTEND      | No operation has been issued, because of the frontend<br><br>The counter counts on any cycle when no operations are issued due to the instruction queue being empty.                                                                                                                                                                     |
| 0x0024       | STALL_BACKEND       | No operation has been issued, because of the backend<br><br>The counter counts on any cycle when no operations are issued due to a pipeline stall.                                                                                                                                                                                       |
| 0x0025       | L1D_TLB             | Level 1 data TLB access<br><br>This event counts any load or store operation which accesses the data L1 TLB. If both a load and a store are executed on a cycle, this event counts twice. This event counts regardless of whether the MMU is enabled.                                                                                    |
| 0x0026       | L1I_TLB             | Level 1 instruction TLB access<br><br>This event counts any instruction fetch which accesses the instruction L1 TLB. This event counts regardless of whether the MMU is enabled.                                                                                                                                                         |
| 0x0029       | L3D_CACHE_ALLOCATE  | Attributable level 3 data or unified cache allocation without refill<br>This event counts any full cache line write into the L3 cache which does not cause a linefill, including write-backs from L2 to L3 and full-line writes which do not allocate into L2.<br><br>This event also counts WriteUnique and datafull WriteEvictOrEvict. |
| 0x002A       | L3D_CACHE_REFILL    | Attributable level 3 data or unified cache refill<br>This event counts for any cacheable read transaction returning data from the SCU for which the data source was outside the cluster.                                                                                                                                                 |
| 0x002B       | L3D_CACHE           | Attributable level 3 data or unified cache access<br>This event counts for any cacheable read, write or write-back transaction sent to the SCU.                                                                                                                                                                                          |
| 0x002D       | L2D_TLB_REFILL      | Level 2 data TLB refill<br><br>This event counts on any refill of the L2 TLB, caused by either an instruction or data access.<br><br>This event does not count if the MMU is disabled.                                                                                                                                                   |

| Event number | Mnemonic           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x002F       | L2D_TLB            | <p>Level 2 data TLB access</p> <p>Attributable level 2 unified TLB access.</p> <p>This event counts on any access to the L2 TLB (caused by a refill of any of the L1 TLBs).</p> <p>This event does not count if the MMU is disabled.</p>                                                                                                                                                                                                                                                                  |
| 0x0031       | REMOTE_ACCESS      | <p>Access to another socket in a multi-socket system</p> <p>This event counts any transactions returning data from another socket in a multi-socket system.</p>                                                                                                                                                                                                                                                                                                                                           |
| 0x0034       | DTLB_WALK          | <p>Data TLB access with at least one translation table walk</p> <p>Access to data TLB that caused a translation table walk.</p> <p>This event counts on any data access which causes L2D_TLB_REFILL to count.</p>                                                                                                                                                                                                                                                                                         |
| 0x0035       | ITLB_WALK          | <p>Instruction TLB access with at least one translation table walk</p> <p>Access to instruction TLB that caused a translation table walk.</p> <p>This event counts on any instruction access which causes L2D_TLB_REFILL to count.</p>                                                                                                                                                                                                                                                                    |
| 0x0036       | LL_CACHE_RD        | <p>Last level cache access, read</p> <p>This event counts any cacheable read transaction which returns a data source of 'interconnect cache', 'DRAM', 'remote' or 'inter-cluster peer'.</p>                                                                                                                                                                                                                                                                                                               |
| 0x0037       | LL_CACHE_MISS_RD   | <p>Last Level cache miss read</p> <p>This event counts any cacheable read transaction which returns a data source of 'DRAM', 'remote' or 'inter-cluster peer'.</p>                                                                                                                                                                                                                                                                                                                                        |
| 0x0039       | L1D_CACHE_LMISS_RD | <p>Level 1 data cache long-latency read miss</p> <p>Level 1 data cache access, read.</p> <p>This event counts any load operation or translation table walk access which looks up in the L1 data cache. In particular, any access which could count the L1D_CACHE_REFILL_RD event causes this event to count.</p> <p>The following instructions are not counted:</p> <ul style="list-style-type: none"> <li>• Cache maintenance instructions and prefetches.</li> <li>• Non-cacheable accesses.</li> </ul> |
| 0x003A       | OP_RETIRED         | <p>Micro-operation architecturally executed</p> <p>This event counts each operation counted by OP_SPEC that would be executed in a simple sequential execution of the program.</p>                                                                                                                                                                                                                                                                                                                        |
| 0x003B       | OP_SPEC            | <p>Micro-operation speculatively executed</p> <p>This event counts the number of operations executed by the core, including those that are executed speculatively and would not be executed in a simple sequential execution of the program.</p>                                                                                                                                                                                                                                                          |

| Event number | Mnemonic               | Description                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x003C       | STALL                  | No operation sent for execution<br><br>This event counts every Attributable cycle on which no Attributable instruction or operation was sent for execution on this core.                                                                                                                                                                                                            |
| 0x003D       | STALL_SLOT_BACKEND     | No operation sent for execution on a slot due to the backend<br><br>Counts each slot counted by STALL_SLOT where no Attributable instruction or operation was sent for execution because the backend is unable to accept one of: <ul style="list-style-type: none"> <li>The instruction operation available for the PE on the slot.</li> <li>Any operations on the slot.</li> </ul> |
| 0x003E       | STALL_SLOT_FRONTEND    | No operation sent for execution on a slot due to the frontend<br><br>Counts each slot counted by STALL_SLOT where no Attributable instruction or operation was sent for execution because there was no Attributable instruction or operation available to issue from the PE from the frontend for the slot.                                                                         |
| 0x003F       | STALL_SLOT             | No operation sent for execution on a slot<br><br>The counter counts on each Attributable cycle the number of instruction or operation slots that are not occupied by an instruction or operation Attributable to the PE.                                                                                                                                                            |
| 0x0040       | L1D_CACHE_RD           | Level 1 data cache access, read<br><br>Counts any load operation or translation table walk access which looks up in the L1 data cache. In particular, any access which could count the L1D_CACHE_REFILL_RD event causes this event to count. Cache maintenance instructions and prefetches are not counted. Non-cacheable accesses are not counted.                                 |
| 0x0041       | L1D_CACHE_WR           | Level 1 data cache access, write<br><br>Counts any store operation which looks up in the L1 data cache. In particular, any access which could count the L1D_CACHE_REFILL event causes this event to count. Cache maintenance instructions and prefetches are not counted. Non-cacheable accesses are not counted.                                                                   |
| 0x0044       | L1D_CACHE_REFILL_INNER | Level 1 data cache refill, inner<br><br>This event counts any L1 data cache linefill (as counted by L1D_CACHE_REFILL) which hits in the L2 cache, L3 cache, or another core in the cluster.                                                                                                                                                                                         |
| 0x0045       | L1D_CACHE_REFILL_OUTER | Level 1 data cache refill, outer<br><br>This event counts any L1 data cache linefill (as counted by L1D_CACHE_REFILL) which does not hit in the L2 cache, L3 cache, or another core in the cluster, and instead obtains data from outside the cluster.                                                                                                                              |
| 0x0046       | L1D_CACHE_WB_VICTIM    | Level 1 data cache write-back, victim                                                                                                                                                                                                                                                                                                                                               |
| 0x0047       | L1D_CACHE_WB_CLEAN     | Level 1 data cache write-back, cleaning and coherency                                                                                                                                                                                                                                                                                                                               |
| 0x0048       | L1D_CACHE_INVALID      | Level 1 data cache invalidate                                                                                                                                                                                                                                                                                                                                                       |
| 0x004C       | L1D_TLB_REFILL_RD      | Level 1 data TLB refill, read                                                                                                                                                                                                                                                                                                                                                       |
| 0x004D       | L1D_TLB_REFILL_WR      | Level 1 data TLB refill, write                                                                                                                                                                                                                                                                                                                                                      |
| 0x004E       | L1D_TLB_RD             | Level 1 data TLB access, read                                                                                                                                                                                                                                                                                                                                                       |
| 0x004F       | L1D_TLB_WR             | Level 1 data TLB access, write                                                                                                                                                                                                                                                                                                                                                      |

| Event number | Mnemonic            | Description                                                                                                                                                                                                                                                                                                               |
|--------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0050       | L2D_CACHE_RD        | Level 2 data cache access, read<br><br>This event counts any transaction issued from L1 caches which looks up in the L2 cache, including requests for instructions fetches and MMU table walks. The transaction is counted regardless of the source that generated it in the L1, being a load, store or prefetch request. |
| 0x0051       | L2D_CACHE_WR        | Level 2 data cache access, write<br><br>This event counts any full cache line write into the L2 cache which does not cause a linefill, including write-backs from L1 to L2, full-line writes which do not allocate into L1 and MMU descriptor hardware updates performed in L2.                                           |
| 0x0052       | L2D_CACHE_REFILL_RD | Level 2 data cache refill, read<br><br>This event counts any cacheable transaction generated by a read operation which causes data to be read from outside the L2.<br><br>This event is counted on RXDAT, using the type of request to determine if the refill was for a read (line requested in any state).              |
| 0x0053       | L2D_CACHE_REFILL_WR | Level 2 data cache refill, write<br><br>This event counts any cacheable transaction generated by a store operation which causes data to be read from outside the L2.<br><br>Counted on RXDAT, using the type of request to determine if the refill was for a write (line requested in unique state).                      |
| 0x0056       | L2D_CACHE_WB_VICTIM | Level 2 data cache write-back, victim<br><br>This event counts any datafull write-back operation caused by allocations.                                                                                                                                                                                                   |
| 0x0057       | L2D_CACHE_WB_CLEAN  | Level 2 data cache write-back, cleaning, and coherency<br><br>This event counts any datafull write-back operation caused by cache maintenance operations or external coherency requests.                                                                                                                                  |
| 0x0058       | L2D_CACHE_INVALID   | Level 2 data cache invalidate<br><br>This event counts any cache maintenance operation which causes the invalidation of a line present in the L2 cache.                                                                                                                                                                   |
| 0x005C       | L2D_TLB_REFILL_RD   | Attributable level 2 data or unified TLB refill, read                                                                                                                                                                                                                                                                     |
| 0x005D       | L2D_TLB_REFILL_WR   | Attributable level 2 data or unified TLB refill, write                                                                                                                                                                                                                                                                    |
| 0x005E       | L2D_TLB_RD          | Attributable level 2 data or unified TLB access, read                                                                                                                                                                                                                                                                     |
| 0x005F       | L2D_TLB_WR          | Attributable level 2 data or unified TLB access, write                                                                                                                                                                                                                                                                    |
| 0x0060       | BUS_ACCESS_RD       | Bus access, read<br><br>This event counts for every beat of data that is transferred over the read data channel between the core and the SCU.                                                                                                                                                                             |
| 0x0061       | BUS_ACCESS_WR       | Bus access, write<br><br>This event counts for every beat of data that is transferred over the write data channel between the core and the SCU.                                                                                                                                                                           |

| Event number | Mnemonic            | Description                                                                                                                                                                                                                                                                                                                               |
|--------------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0066       | MEM_ACCESS_RD       | Data memory access, read<br><br>This event counts memory accesses due to load instructions. The following instructions are not counted:<br><ul style="list-style-type: none"> <li>The following instructions are not counted: - Instruction fetches - Cache maintenance instructions. - Translation table walks. - Prefetches.</li> </ul> |
| 0x0067       | MEM_ACCESS_WR       | Data memory access, write<br><br>This event counts memory accesses due to store instructions.<br><br>The following instructions are not counted:<br><ul style="list-style-type: none"> <li>Instruction fetches.</li> <li>Cache maintenance instructions.</li> <li>Translation table walks.</li> <li>Prefetches.</li> </ul>                |
| 0x0068       | UNALIGNED_LD_SPEC   | Unaligned access, read                                                                                                                                                                                                                                                                                                                    |
| 0x0069       | UNALIGNED_ST_SPEC   | Unaligned access, write                                                                                                                                                                                                                                                                                                                   |
| 0x006A       | UNALIGNED_LDST_SPEC | Unaligned access                                                                                                                                                                                                                                                                                                                          |
| 0x006C       | LDREX_SPEC          | Exclusive operation speculatively executed, LDREX or LDX                                                                                                                                                                                                                                                                                  |
| 0x006D       | STREX_PASS_SPEC     | Exclusive operation speculatively executed, STREX or STX pass                                                                                                                                                                                                                                                                             |
| 0x006E       | STREX_FAIL_SPEC     | Exclusive operation speculatively executed, Store-Exclusive fail.<br><br>Exclusive operation speculatively executed, STREX or STX fail.                                                                                                                                                                                                   |
| 0x006F       | STREX_SPEC          | Exclusive operation speculatively executed, Store-Exclusive.<br><br>Exclusive operation speculatively executed, STREX or STX.                                                                                                                                                                                                             |
| 0x0070       | LD_SPEC             | Operation speculatively executed, load                                                                                                                                                                                                                                                                                                    |
| 0x0071       | ST_SPEC             | Operation speculatively executed, store                                                                                                                                                                                                                                                                                                   |
| 0x0072       | LDST_SPEC           | Operation speculatively executed, load or store                                                                                                                                                                                                                                                                                           |
| 0x0073       | DP_SPEC             | Operation speculatively executed, integer data processing                                                                                                                                                                                                                                                                                 |
| 0x0074       | ASE_SPEC            | Operation speculatively executed, Advanced SIMD                                                                                                                                                                                                                                                                                           |
| 0x0075       | VFP_SPEC            | Operation speculatively executed, floating-point                                                                                                                                                                                                                                                                                          |
| 0x0076       | PC_WRITE_SPEC       | Operation speculatively executed, software change of the PC                                                                                                                                                                                                                                                                               |
| 0x0077       | CRYPTO_SPEC         | Operation speculatively executed, Cryptographic instruction                                                                                                                                                                                                                                                                               |
| 0x0078       | BR_IMMED_SPEC       | Branch speculatively executed, immediate branch                                                                                                                                                                                                                                                                                           |
| 0x0079       | BR_RETURN_SPEC      | Branch speculatively executed, procedure return                                                                                                                                                                                                                                                                                           |
| 0x007A       | BR_INDIRECT_SPEC    | Branch speculatively executed, indirect branch                                                                                                                                                                                                                                                                                            |
| 0x007C       | ISB_SPEC            | Barrier speculatively executed, ISB                                                                                                                                                                                                                                                                                                       |
| 0x007D       | DSB_SPEC            | Barrier speculatively executed, DSB                                                                                                                                                                                                                                                                                                       |
| 0x007E       | DMB_SPEC            | Barrier speculatively executed, DMB                                                                                                                                                                                                                                                                                                       |
| 0x0081       | EXC_UNDEF           | Exception taken, other synchronous<br><br>Counts the number of undefined exceptions taken locally                                                                                                                                                                                                                                         |

| Event number | Mnemonic        | Description                                                                                                                                                                                                                                                                                                                                                      |
|--------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0082       | EXC_SVC         | Exception taken, Supervisor Call<br>Exception taken locally, Supervisor Call                                                                                                                                                                                                                                                                                     |
| 0x0083       | EXC_PABORT      | Exception taken, Instruction Abort<br>Exception taken locally, Instruction Abort                                                                                                                                                                                                                                                                                 |
| 0x0084       | EXC_DABORT      | Exception taken, Data Abort or SError<br>Exception taken locally, Data Abort and SError                                                                                                                                                                                                                                                                          |
| 0x0086       | EXC_IRQ         | Exception taken, IRQ<br>Exception taken locally, IRQ                                                                                                                                                                                                                                                                                                             |
| 0x0087       | EXC_FIQ         | Exception taken, FIQ<br>Exception taken locally, FIQ                                                                                                                                                                                                                                                                                                             |
| 0x0088       | EXC_SMC         | Exception taken, Secure Monitor Call<br>Exception taken locally, Secure Monitor Call                                                                                                                                                                                                                                                                             |
| 0x008A       | EXC_HVC         | Exception taken, Hypervisor Call<br>Exception taken locally, Hypervisor Call                                                                                                                                                                                                                                                                                     |
| 0x008B       | EXC_TRAP_PABORT | Exception taken, Instruction Abort not Taken locally                                                                                                                                                                                                                                                                                                             |
| 0x008C       | EXC_TRAP_DABORT | Exception taken, Data Abort or SError not Taken locally                                                                                                                                                                                                                                                                                                          |
| 0x008D       | EXC_TRAP_OTHER  | Exception taken, other traps not Taken locally                                                                                                                                                                                                                                                                                                                   |
| 0x008E       | EXC_TRAP_IRQ    | Exception taken, IRQ not Taken locally                                                                                                                                                                                                                                                                                                                           |
| 0x008F       | EXC_TRAP_FIQ    | Exception taken, FIQ not Taken locally                                                                                                                                                                                                                                                                                                                           |
| 0x0090       | RC_LD_SPEC      | Release consistency operation speculatively executed, Load-Acquire                                                                                                                                                                                                                                                                                               |
| 0x0091       | RC_ST_SPEC      | Release consistency operation speculatively executed, Store-Release                                                                                                                                                                                                                                                                                              |
| 0x00A0       | L3D_CACHE_RD    | Attributable level 3 data or unified cache access, read<br>This event counts for any cacheable read transaction sent to the SCU.                                                                                                                                                                                                                                 |
| 0x4000       | SAMPLE_POP      | Statistical Profiling sample population<br>The counter increments for each operation that might be sampled, whether or not the operation was sampled. Operations that are executed at an Exception level or Security state in which the Statistical Profiling Extension is disabled are not counted.                                                             |
| 0x4001       | SAMPLE_FEED     | Statistical Profiling sample taken<br>The counter increments each time the sample interval counter reaches zero and is reloaded, and the sample does not collide with the previous sample. Samples that are removed by filtering, or discarded, and not written to the Profiling Buffer are counted.                                                             |
| 0x4002       | SAMPLE_FILTRATE | Statistical Profiling sample taken and not removed by filtering<br>The counter increments each time that a completed sample record is checked against the filters and not removed. Sample records that are not removed by filtering, but are discarded before being written to the Profiling Buffer because of a Profiling Buffer management event, are counted. |



| Event number | Mnemonic           | Description                                                                                                                                                                                                                        |
|--------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x4003       | SAMPLE_COLLISION   | Statistical Profiling sample collided with previous sample<br><br>The counter increments for each sample record that is taken when the previous sampled operation has not completed generating its sample record.                  |
| 0x4004       | CNT_CYCLES         | Constant frequency cycles                                                                                                                                                                                                          |
| 0x4005       | STALL_BACKEND_MEM  | Memory stall cycles<br><br>The counter is defined identically to STALL_BACKEND_MEM in the AMUv1 architecture.                                                                                                                      |
| 0x4006       | L1I_CACHE_LMISS    | Level 1 instruction cache long-latency miss<br><br>The counter counts each access counted by L1I_CACHE that incurs additional latency because it returns instructions from outside the Level 1 instruction cache.                  |
| 0x4009       | L2D_CACHE_LMISS_RD | Level 2 data cache long-latency read miss<br><br>The counter counts each memory read access counted by L2D_CACHE that incurs additional latency because it returns data from outside the Level 2 data or unified cache of this PE. |
| 0x400B       | L3D_CACHE_LMISS_RD | Level 3 data cache long-latency read miss<br><br>The counter counts each memory read access counted by L3D_CACHE that incurs additional latency because it returns data from outside the Level 3 data or unified cache of this PE. |
| 0x400C       | TRB_WRAP           | Trace buffer current write pointer wrapped                                                                                                                                                                                         |
| 0x400D       | PMU_OVFS           | PMU overflow, counters accessible to EL1 and EL0<br><br><b>Note:</b><br>This event is exported to the trace unit, but cannot be counted in the PMU.                                                                                |
| 0x400E       | TRB_TRIG           | Trace buffer Trigger Event<br><br><b>Note:</b><br>This event is exported to the trace unit, but cannot be counted in the PMU.                                                                                                      |
| 0x400F       | PMU_HOVFS          | PMU overflow, counters reserved for use by EL2<br><br><b>Note:</b><br>This event is exported to the trace unit, but cannot be counted in the PMU.                                                                                  |
| 0x4010       | TRCEXTOUT0         | PE Trace Unit external output 0<br><br><b>Note:</b><br>This event is not exported to the trace unit.                                                                                                                               |
| 0x4011       | TRCEXTOUT1         | PE Trace Unit external output 1<br><br><b>Note:</b><br>This event is not exported to the trace unit.                                                                                                                               |

| Event number | Mnemonic               | Description                                                                                                                                                                                                       |
|--------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x4012       | TRCEXTOUT2             | PE Trace Unit external output 2<br><br><b>Note:</b><br>This event is not exported to the trace unit.                                                                                                              |
| 0x4013       | TRCEXTOUT3             | PE Trace Unit external output 3<br><br><b>Note:</b><br>This event is not exported to the trace unit.                                                                                                              |
| 0x4018       | CTI_TRIGOUT4           | Cross Trigger Interface output trigger 4                                                                                                                                                                          |
| 0x4019       | CTI_TRIGOUT5           | Cross Trigger Interface output trigger 5                                                                                                                                                                          |
| 0x401A       | CTI_TRIGOUT6           | Cross Trigger Interface output trigger 6                                                                                                                                                                          |
| 0x401B       | CTI_TRIGOUT7           | Cross Trigger Interface output trigger 7                                                                                                                                                                          |
| 0x4020       | LDST_ALIGN_LAT         | Access with additional latency from alignment<br><br>The counter counts each access counted by MEM_ACCESS that, due to the alignment of the address and size of data being accessed, incurred additional latency. |
| 0x4021       | LD_ALIGN_LAT           | Load with additional latency from alignment                                                                                                                                                                       |
| 0x4022       | ST_ALIGN_LAT           | Store with additional latency from alignment                                                                                                                                                                      |
| 0x4024       | MEM_ACCESS_CHECKED     | Checked data memory access                                                                                                                                                                                        |
| 0x4025       | MEM_ACCESS_RD_CHECKED  | Checked data memory access, read                                                                                                                                                                                  |
| 0x4026       | MEM_ACCESS_WR_CHECKED  | Checked data memory access, write                                                                                                                                                                                 |
| 0x8004       | SIMD_INST_SPEC         | SIMD instruction speculatively executed                                                                                                                                                                           |
| 0x8005       | ASE_INST_SPEC          | Advanced SIMD operations speculatively executed                                                                                                                                                                   |
| 0x8006       | SVE_INST_SPEC          | SVE operation, including load/store                                                                                                                                                                               |
| 0x8014       | FP_HP_SPEC             | Half-precision floating-point operation speculatively executed                                                                                                                                                    |
| 0x8018       | FP_SP_SPEC             | Single-precision floating-point operation speculatively executed                                                                                                                                                  |
| 0x801C       | FP_DP_SPEC             | Double-precision floating-point operation speculatively executed                                                                                                                                                  |
| 0x8040       | INT_SPEC               | Advanced SIMD and SVE integer operations speculatively executed                                                                                                                                                   |
| 0x8074       | SVE_PRED_SPEC          | SVE predicated operations speculatively executed                                                                                                                                                                  |
| 0x8075       | SVE_PRED_EMPTY_SPEC    | SVE predicated operations with no active predicates speculatively executed                                                                                                                                        |
| 0x8076       | SVE_PRED_FULL_SPEC     | SVE predicated operations with all active predicates speculatively executed                                                                                                                                       |
| 0x8077       | SVE_PRED_PARTIAL_SPEC  | SVE predicated operations with partially active predicates speculatively executed                                                                                                                                 |
| 0x8079       | SVE_PRED_NOT_FULL_SPEC | SVE predicated operations with no or partially active predicates speculatively executed                                                                                                                           |
| 0x80BC       | SVE_LDFF_SPEC          | SVE First-fault load operations speculatively executed                                                                                                                                                            |
| 0x80BD       | SVE_LDFF_FAULT_SPEC    | SVE First-fault load operations speculatively executed which set FFR bit to 0                                                                                                                                     |
| 0x80C0       | FP_SCALE_OPS_SPEC      | Scalable floating-point element operations speculatively executed                                                                                                                                                 |
| 0x80C1       | FP_FIXED_OPS_SPEC      | Non-scalable floating-point element operations speculatively executed                                                                                                                                             |
| 0x80E3       | ASE_SVE_INT8_SPEC      | Advanced SIMD and SVE 8-bit integer operation speculatively executed                                                                                                                                              |
| 0x80E7       | ASE_SVE_INT16_SPEC     | Advanced SIMD and SVE 16-bit integer operation speculatively executed                                                                                                                                             |
| 0x80EB       | ASE_SVE_INT32_SPEC     | Advanced SIMD and SVE 32-bit integer operation speculatively executed                                                                                                                                             |

| Event number | Mnemonic                   | Description                                                                                                                         |
|--------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 0x80EF       | ASE_SVE_INT64_SPEC         | Advanced SIMD and SVE 64-bit integer operation speculatively executed                                                               |
| 0x8087       | PRF_SPEC                   | The counter counts speculatively executed prefetch operations due to scalar PRFM and SVE PRFInstructions                            |
| 0x810C       | BR_INDNR_TAKEN_RETIRED     | Instruction architecturally executed, indirect branch taken excluding returns                                                       |
| 0x8110       | BR_IMMED_PRED_RETIRED      | Instruction architecturally executed, predicted immediate branch                                                                    |
| 0x8111       | BR_IMMED_MIS_PRED_RETIRED  | Instruction architecturally executed, mispredicted immediate branch                                                                 |
| 0x8112       | BR_IND_PRED_RETIRED        | Instruction architecturally executed, predicted indirect branch                                                                     |
| 0x8113       | BR_IND_MIS_PRED_RETIRED    | Instruction architecturally executed, mispredicted indirect branch                                                                  |
| 0x8114       | BR_RETURN_PRED_RETIRED     | Instruction architecturally executed, predicted procedure return                                                                    |
| 0x8115       | BR_RETURN_MIS_PRED_RETIRED | Instruction architecturally executed, mispredicted procedure return                                                                 |
| 0x8116       | BR_INDNR_PRED_RETIRED      | Instruction architecturally executed, predicted indirect branch excluding returns                                                   |
| 0x8117       | BR_INDNR_MIS_PRED_RETIRED  | Instruction architecturally executed, mispredicted indirect branch excluding returns                                                |
| 0x8118       | BR_TAKEN_PRED_RETIRED      | Instruction architecturally executed, predicted taken branch                                                                        |
| 0x8119       | BR_TAKEN_MIS_PRED_RETIRED  | Instruction architecturally executed, mispredicted taken branch                                                                     |
| 0x811A       | BR_SKIP_PRED_RETIRED       | Instruction architecturally executed, predicted not taken branch                                                                    |
| 0x811B       | BR_SKIP_MIS_PRED_RETIRED   | Instruction architecturally executed, mispredicted not taken branch                                                                 |
| 0x811C       | BR_PRED_RETIRED            | Instruction architecturally executed, predicted branch                                                                              |
| 0x811D       | BR_IND_RETIRED             | Instruction architecturally executed, indirect branch                                                                               |
| 0x8120       | INST_FETCH_PERCYC          | Total cycles, INST_FETCH                                                                                                            |
| 0x8121       | MEM_ACCESS_RD_PERCYC       | Total cycles, MEM_ACCESS_RD                                                                                                         |
| 0x8124       | INST_FETCH                 | Instruction memory access                                                                                                           |
| 0x8128       | DTLB_WALK_PERCYC           | Total cycles, DTLB_WALK<br><br>The counter counts by the number of data TLB walk events in progress on each processor cycle.        |
| 0x8129       | ITLB_WALK_PERCYC           | Total cycles, ITLB_WALK<br><br>The counter counts by the number of instruction TLB walk events in progress on each processor cycle. |
| 0x812A       | SAMPLE_FEED_BR             | Statistical Profiling sample taken, branch                                                                                          |
| 0x812B       | SAMPLE_FEED_LD             | Statistical Profiling sample taken, load                                                                                            |
| 0x812C       | SAMPLE_FEED_ST             | Statistical Profiling sample taken, store                                                                                           |
| 0x812D       | SAMPLE_FEED_OP             | Statistical Profiling sample taken, matching operation type                                                                         |
| 0x812E       | SAMPLE_FEED_EVENT          | Statistical Profiling sample taken, matching events                                                                                 |
| 0x812F       | SAMPLE_FEED_LAT            | Statistical Profiling sample taken, exceeding minimum latency                                                                       |
| 0x8130       | L1D_TLB_RW                 | Level 1 data or unified TLB demand access                                                                                           |
| 0x8131       | L1I_TLB_RD                 | Level 1 instruction TLB demand access                                                                                               |
| 0x8132       | L1D_TLB_PRFM               | Level 1 data or unified TLB preload or prefetch                                                                                     |
| 0x8133       | L1I_TLB_PRFM               | Level 1 instruction TLB preload or prefetch                                                                                         |
| 0x8134       | DTLB_HWUPD                 | Data TLB hardware update of translation table                                                                                       |
| 0x8135       | ITLB_HWUPD                 | Instruction TLB hardware update of translation table                                                                                |
| 0x8136       | DTLB_STEP                  | Data TLB translation table walk, step                                                                                               |

| Event number | Mnemonic                | Description                                                                      |
|--------------|-------------------------|----------------------------------------------------------------------------------|
| 0x8137       | ITLB_STEP               | Instruction TLB translation table walk, step                                     |
| 0x8138       | DTLB_WALK_LARGE         | Data TLB large page translation table walk                                       |
| 0x8139       | ITLB_WALK_LARGE         | Instruction TLB large page translation table walk                                |
| 0x813A       | DTLB_WALK_SMALL         | Data TLB small page translation table walk                                       |
| 0x813B       | ITLB_WALK_SMALL         | Instruction TLB small page translation table walk                                |
| 0x813C       | DTLB_WALK_RW            | Data TLB demand access with at least one translation table walk                  |
| 0x813D       | ITLB_WALK_RD            | Instruction TLB demand access with at least one translation table walk           |
| 0x813E       | DTLB_WALK_PRFM          | Data TLB preload or prefetch with at least one translation table walk            |
| 0x813F       | ITLB_WALK_PRFM          | Instruction TLB preload or prefetch with at least one translation table walk     |
| 0x81C0       | L1I_CACHE_HIT_RD        | Level 1 instruction cache demand fetch hit                                       |
| 0x81D0       | L1I_CACHE_HIT_RD_FPRFM  | Level 1 instruction cache demand fetch first hit, fetched by software preload    |
| 0x81E0       | L1I_CACHE_HIT_RD_FHWPRF | Level 1 instruction cache demand fetch first hit, fetched by hardware prefetcher |
| 0x8140       | L1D_CACHE_RW            | Level 1 data cache demand access                                                 |
| 0x8141       | L1I_CACHE_RD            | Level 1 instruction cache demand access                                          |
| 0x8142       | L1D_CACHE_PRFM          | Level 1 data cache preload or prefetch                                           |
| 0x8143       | L1I_CACHE_PRFM          | Level 1 instruction cache preload or prefetch                                    |
| 0x8144       | L1D_CACHE_MISS          | Level 1 data cache demand access miss                                            |
| 0x8145       | L1I_CACHE_HWPRF         | Level 1 instruction cache hardware prefetch                                      |
| 0x8146       | L1D_CACHE_REFILL_PRFM   | Level 1 data cache refill, preload or prefetch                                   |
| 0x8147       | L1I_CACHE_REFILL_PRFM   | Level 1 instruction cache refill,preload or prefetch                             |
| 0x8148       | L2D_CACHE_RW            | Level 2 data cache demand access                                                 |
| 0x814A       | L2D_CACHE_PRFM          | Level 2 data cache preload or prefetch                                           |
| 0x814C       | L2D_CACHE_MISS          | Level 2 data cache demand access miss                                            |
| 0x814E       | L2D_CACHE_REFILL_PRFM   | Level 2 data cache refill, preload or prefetch                                   |
| 0x8150       | L3D_CACHE_RW            | Level 3 data cache demand access                                                 |
| 0x8151       | L3D_CACHE_PRFM          | Level 3 data cache preload or prefetch                                           |
| 0x8152       | L3D_CACHE_MISS          | Level 3 data cache demand access miss                                            |
| 0x8153       | L3D_CACHE_REFILL_PRFM   | Level 3 data cache refill,preload or prefetch                                    |
| 0x8154       | L1D_CACHE_HWPRF         | Level 1 data cache hardware prefetch                                             |
| 0x8155       | L2D_CACHE_HWPRF         | Level 2 data cache hardware prefetch                                             |
| 0x8158       | STALL_FRONTEND_MEMBOUND | Frontend stall cycles, memory bound                                              |
| 0x8159       | STALL_FRONTEND_L1I      | Frontend stall cycles, level 1 instruction cache                                 |
| 0x815B       | STALL_FRONTEND_MEM      | Frontend stall cycles, last level PE cache or memory                             |
| 0x815C       | STALL_FRONTEND_TLB      | Frontend stall cycles, TLB                                                       |
| 0x8160       | STALL_FRONTEND_CPUBOUND | Frontend stall cycles, processor bound                                           |
| 0x8161       | STALL_FRONTEND_FLOW     | Frontend stall cycles, flow control                                              |
| 0x8162       | STALL_FRONTEND_FLUSH    | Frontend stall cycles, flush recovery                                            |
| 0x8164       | STALL_BACKEND_MEMBOUND  | Backend stall cycles, memory bound                                               |
| 0x8165       | STALL_BACKEND_L1D       | Backend stall cycles, level 1 data cache                                         |
| 0x8167       | STALL_BACKEND_TLB       | Backend stall cycles, TLB                                                        |

| Event number | Mnemonic               | Description                                                                                                |
|--------------|------------------------|------------------------------------------------------------------------------------------------------------|
| 0x8168       | STALL_BACKEND_ST       | Backend stall cycles, store                                                                                |
| 0x816A       | STALL_BACKEND_CPUBOUND | Backend stall cycles, processor bound                                                                      |
| 0x816B       | STALL_BACKEND_BUSY     | Backend stall cycles, backend busy                                                                         |
| 0x816D       | STALL_BACKEND_RENAME   | Backend stall cycles, rename full                                                                          |
| 0x8200       | L1I_CACHE_HIT          | Level 1 instruction cache hit                                                                              |
| 0x8208       | L1I_CACHE_HIT_PRFM     | Level 1 instruction cache software preload hit                                                             |
| 0x8240       | L1I_LFB_HIT_RD         | Level 1 instruction cache demand fetch line-fill buffer hit                                                |
| 0x8250       | L1I_LFB_HIT_RD_FPRFM   | Level 1 instruction cache demand fetch line-fill buffer first hit, recently fetched by software preload    |
| 0x8260       | L1I_LFB_HIT_RD_FHWPRF  | Level 1 instruction cache demand fetch line-fill buffer first hit, recently fetched by hardware prefetcher |

## 17.2 Performance monitors interrupts

The *Performance Monitoring Unit* (PMU) can be configured to generate an interrupt when one or more of the counters overflow.

When the PMU generates an interrupt, the nPMUIRQ[n] output is driven LOW.

See *Performance Monitors Extension support* in the [Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual](#) for more information.

## 17.3 External register access permissions

The Cortex®-X925 core supports access to the *Performance Monitoring Unit* (PMU) registers from the system register interface and a memory-mapped interface.

Access to a register depends on:

- Whether the core is powered up
- The state of the OS Lock
- The state of External Performance Monitors Access Disable

The behavior is specific to each register and is not described in this manual. For a detailed description of these features and their effects on the registers, see the [Arm® Architecture Reference Manual for A-profile architecture](#). The register descriptions provided in this manual describe whether each register is read/write or read-only.

## 17.4 AArch64 Performance Monitors registers

The following summary table provides an overview of all Performance Monitors registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 17-2: Performance Monitors registers summary**

| Name                        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                 |
|-----------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------------|
| PMINTENSET_EL1              | 3   | 0   | C9  | C14 | 1   | See individual bit resets. | 64-bit | Performance Monitors Interrupt Enable Set Register          |
| PMINTENCLR_EL1              | 3   | 0   | C9  | C14 | 2   | See individual bit resets. | 64-bit | Performance Monitors Interrupt Enable Clear Register        |
| <a href="#">PMMIR_EL1</a>   | 3   | 0   | C9  | C14 | 6   | See individual bit resets. | 64-bit | Performance Monitors Machine Identification Register        |
| <a href="#">PMCR_ELO</a>    | 3   | 3   | C9  | C12 | 0   | See individual bit resets. | 64-bit | Performance Monitors Control Register                       |
| PMCNTENSET_ELO              | 3   | 3   | C9  | C12 | 1   | See individual bit resets. | 64-bit | Performance Monitors Count Enable Set Register              |
| PMCNTENCLR_ELO              | 3   | 3   | C9  | C12 | 2   | See individual bit resets. | 64-bit | Performance Monitors Count Enable Clear Register            |
| PMOVSLR_ELO                 | 3   | 3   | C9  | C12 | 3   | See individual bit resets. | 64-bit | Performance Monitors Overflow Flag Status Clear Register    |
| PMSWINC_ELO                 | 3   | 3   | C9  | C12 | 4   | See individual bit resets. | 64-bit | Performance Monitors Software Increment Register            |
| PMSELR_ELO                  | 3   | 3   | C9  | C12 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Counter Selection Register       |
| <a href="#">PMCEID0_ELO</a> | 3   | 3   | C9  | C12 | 6   | See individual bit resets. | 64-bit | Performance Monitors Common Event Identification Register 0 |
| <a href="#">PMCEID1_ELO</a> | 3   | 3   | C9  | C12 | 7   | See individual bit resets. | 64-bit | Performance Monitors Common Event Identification Register 1 |
| PMCCNTR_ELO                 | 3   | 3   | C9  | C13 | 0   | See individual bit resets. | 64-bit | Performance Monitors Cycle Count Register                   |
| PMXEVTYPER_ELO              | 3   | 3   | C9  | C13 | 1   | See individual bit resets. | 64-bit | Performance Monitors Selected Event Type Register           |
| PMXVCNTR_ELO                | 3   | 3   | C9  | C13 | 2   | See individual bit resets. | 64-bit | Performance Monitors Selected Event Count Register          |
| PMUSERENR_ELO               | 3   | 3   | C9  | C14 | 0   | See individual bit resets. | 64-bit | Performance Monitors User Enable Register                   |
| PMOVSSET_ELO                | 3   | 3   | C9  | C14 | 3   | See individual bit resets. | 64-bit | Performance Monitors Overflow Flag Status Set Register      |

| Name           | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| PMEVCNTR0_ELO  | 3   | 3   | C14 | C8  | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR1_ELO  | 3   | 3   | C14 | C8  | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR2_ELO  | 3   | 3   | C14 | C8  | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR3_ELO  | 3   | 3   | C14 | C8  | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR4_ELO  | 3   | 3   | C14 | C8  | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR5_ELO  | 3   | 3   | C14 | C8  | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR6_ELO  | 3   | 3   | C14 | C8  | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR7_ELO  | 3   | 3   | C14 | C8  | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR8_ELO  | 3   | 3   | C14 | C9  | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR9_ELO  | 3   | 3   | C14 | C9  | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR10_ELO | 3   | 3   | C14 | C9  | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR11_ELO | 3   | 3   | C14 | C9  | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR12_ELO | 3   | 3   | C14 | C9  | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR13_ELO | 3   | 3   | C14 | C9  | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR14_ELO | 3   | 3   | C14 | C9  | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR15_ELO | 3   | 3   | C14 | C9  | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR16_ELO | 3   | 3   | C14 | C10 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR17_ELO | 3   | 3   | C14 | C10 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR18_ELO | 3   | 3   | C14 | C10 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR19_ELO | 3   | 3   | C14 | C10 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR20_ELO | 3   | 3   | C14 | C10 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR21_ELO | 3   | 3   | C14 | C10 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR22_ELO | 3   | 3   | C14 | C10 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| PMEVCNTR23_ELO  | 3   | 3   | C14 | C10 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR24_ELO  | 3   | 3   | C14 | C11 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR25_ELO  | 3   | 3   | C14 | C11 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR26_ELO  | 3   | 3   | C14 | C11 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR27_ELO  | 3   | 3   | C14 | C11 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR28_ELO  | 3   | 3   | C14 | C11 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR29_ELO  | 3   | 3   | C14 | C11 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR30_ELO  | 3   | 3   | C14 | C11 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVTYPER0_ELO  | 3   | 3   | C14 | C12 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER1_ELO  | 3   | 3   | C14 | C12 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER2_ELO  | 3   | 3   | C14 | C12 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER3_ELO  | 3   | 3   | C14 | C12 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER4_ELO  | 3   | 3   | C14 | C12 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER5_ELO  | 3   | 3   | C14 | C12 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER6_ELO  | 3   | 3   | C14 | C12 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER7_ELO  | 3   | 3   | C14 | C12 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER8_ELO  | 3   | 3   | C14 | C13 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER9_ELO  | 3   | 3   | C14 | C13 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER10_ELO | 3   | 3   | C14 | C13 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER11_ELO | 3   | 3   | C14 | C13 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER12_ELO | 3   | 3   | C14 | C13 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER13_ELO | 3   | 3   | C14 | C13 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER14_ELO | 3   | 3   | C14 | C13 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |



| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                      |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------|
| PMEVTYPER15_ELO | 3   | 3   | C14 | C13 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER16_ELO | 3   | 3   | C14 | C14 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER17_ELO | 3   | 3   | C14 | C14 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER18_ELO | 3   | 3   | C14 | C14 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER19_ELO | 3   | 3   | C14 | C14 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER20_ELO | 3   | 3   | C14 | C14 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER21_ELO | 3   | 3   | C14 | C14 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER22_ELO | 3   | 3   | C14 | C14 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER23_ELO | 3   | 3   | C14 | C14 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER24_ELO | 3   | 3   | C14 | C15 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER25_ELO | 3   | 3   | C14 | C15 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER26_ELO | 3   | 3   | C14 | C15 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER27_ELO | 3   | 3   | C14 | C15 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER28_ELO | 3   | 3   | C14 | C15 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER29_ELO | 3   | 3   | C14 | C15 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER30_ELO | 3   | 3   | C14 | C15 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMCCFILTR_ELO   | 3   | 3   | C14 | C15 | 7   | See individual bit resets. | 64-bit | Performance Monitors Cycle Count Filter Register |

## 17.5 External PMU registers summary

The following summary table provides an overview of all memory-mapped PMU registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 17-3: PMU registers summary**

| Offset | Name                  | Reset                      | Width  | Description                                |
|--------|-----------------------|----------------------------|--------|--------------------------------------------|
| 0x0    | PMEVCNTR0_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x8    | PMEVCNTR1_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x10   | PMEVCNTR2_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x18   | PMEVCNTR3_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x20   | PMEVCNTR4_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x28   | PMEVCNTR5_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x30   | PMEVCNTR6_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x38   | PMEVCNTR7_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x40   | PMEVCNTR8_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x48   | PMEVCNTR9_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x50   | PMEVCNTR10_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x58   | PMEVCNTR11_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x60   | PMEVCNTR12_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x68   | PMEVCNTR13_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x70   | PMEVCNTR14_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x78   | PMEVCNTR15_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x80   | PMEVCNTR16_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x88   | PMEVCNTR17_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x90   | PMEVCNTR18_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x98   | PMEVCNTR19_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xA0   | PMEVCNTR20_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xA8   | PMEVCNTR21_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xB0   | PMEVCNTR22_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xB8   | PMEVCNTR23_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xC0   | PMEVCNTR24_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xC8   | PMEVCNTR25_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xD0   | PMEVCNTR26_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xD8   | PMEVCNTR27_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xE0   | PMEVCNTR28_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xE8   | PMEVCNTR29_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xF0   | PMEVCNTR30_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x0F8  | PMCCNTR_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Cycle Counter         |
| 0x0FC  | PMCCNTR_ELO [63:32]   | See individual bit resets. | 32-bit | Performance Monitors Cycle Counter         |
| 0x200  | PMPCSR [31:0]         | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x204  | PMPCSR [63:32]        | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x220  | PMPCSR [31:0]         | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x224  | PMPCSR [63:32]        | See individual bit resets. | 32-bit | Program Counter Sample Register            |

| Offset | Name                      | Reset                      | Width  | Description                                        |
|--------|---------------------------|----------------------------|--------|----------------------------------------------------|
| 0x208  | PMCID1SR                  | See individual bit resets. | 32-bit | CONTEXTIDR_EL1 Sample Register                     |
| 0x228  | PMCID1SR                  | See individual bit resets. | 32-bit | CONTEXTIDR_EL1 Sample Register                     |
| 0x20C  | PMVIDSR                   | See individual bit resets. | 32-bit | VMID Sample Register                               |
| 0x22C  | PMCID2SR                  | See individual bit resets. | 32-bit | CONTEXTIDR_EL2 Sample Register                     |
| 0x400  | PMEVTYPER0_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x404  | PMEVTYPER1_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x408  | PMEVTYPER2_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x40C  | PMEVTYPER3_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x410  | PMEVTYPER4_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x414  | PMEVTYPER5_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x418  | PMEVTYPER6_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x41C  | PMEVTYPER7_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x420  | PMEVTYPER8_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x424  | PMEVTYPER9_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x428  | PMEVTYPER10_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x42C  | PMEVTYPER11_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x430  | PMEVTYPER12_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x434  | PMEVTYPER13_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x438  | PMEVTYPER14_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x43C  | PMEVTYPER15_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x440  | PMEVTYPER16_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x444  | PMEVTYPER17_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x448  | PMEVTYPER18_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x44C  | PMEVTYPER19_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x450  | PMEVTYPER20_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x454  | PMEVTYPER21_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x458  | PMEVTYPER22_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x45C  | PMEVTYPER23_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x460  | PMEVTYPER24_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x464  | PMEVTYPER25_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x468  | PMEVTYPER26_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x46C  | PMEVTYPER27_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x470  | PMEVTYPER28_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x474  | PMEVTYPER29_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x478  | PMEVTYPER30_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x47C  | PMCCFILTR_ELO [31:0]      | See individual bit resets. | 32-bit | Performance Monitors Cycle Counter Filter Register |
| 0x600  | <a href="#">PMPCSSR</a>   | See individual bit resets. | 64-bit | Snapshot Program Counter Sample Register           |
| 0x608  | <a href="#">PMCIDSSR</a>  | See individual bit resets. | 32-bit | Snapshot CONTEXTIDR_EL1 Sample Register            |
| 0x60C  | <a href="#">PMCID2SSR</a> | See individual bit resets. | 32-bit | Snapshot CONTEXTIDR_EL2 Sample Register            |
| 0x610  | <a href="#">PMSSSR</a>    | See individual bit resets. | 32-bit | PMU Snapshot Status Register                       |

| Offset | Name                          | Reset                      | Width  | Description                                              |
|--------|-------------------------------|----------------------------|--------|----------------------------------------------------------|
| 0x618  | <a href="#">PMCCNTSR</a>      | See individual bit resets. | 64-bit | PMU Cycle Counter Snapshot Register                      |
| 0x620  | PMEVCNTRS0                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x628  | PMEVCNTRS1                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x630  | PMEVCNTRS2                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x638  | PMEVCNTRS3                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x640  | PMEVCNTRS4                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x648  | PMEVCNTRS5                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x650  | PMEVCNTRS6                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x658  | PMEVCNTRS7                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x660  | PMEVCNTRS8                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x668  | PMEVCNTRS9                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x670  | PMEVCNTRS10                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x678  | PMEVCNTRS11                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x680  | PMEVCNTRS12                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x688  | PMEVCNTRS13                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x690  | PMEVCNTRS14                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x698  | PMEVCNTRS15                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6A0  | PMEVCNTRS16                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6A8  | PMEVCNTRS17                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6B0  | PMEVCNTRS18                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6B8  | PMEVCNTRS19                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6C0  | PMEVCNTRS20                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6C8  | PMEVCNTRS21                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6D0  | PMEVCNTRS22                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6D8  | PMEVCNTRS23                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6E0  | PMEVCNTRS24                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6E8  | PMEVCNTRS25                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6F0  | PMEVCNTRS26                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x6F8  | PMEVCNTRS27                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x700  | PMEVCNTRS28                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x708  | PMEVCNTRS29                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0x710  | PMEVCNTRS30                   | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                      |
| 0xC00  | PMCNTENSET_ELO [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Count Enable Set Register           |
| 0xC20  | PMCNTENCLR_ELO [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Count Enable Clear Register         |
| 0xC40  | PMINTENSET_EL1 [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Interrupt Enable Set Register       |
| 0xC60  | PMINTENCLR_EL1 [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Interrupt Enable Clear Register     |
| 0xC80  | PMOVSLR_ELO [31:0]            | See individual bit resets. | 32-bit | Performance Monitors Overflow Flag Status Clear register |
| 0xCA0  | PMSWINC_ELO                   | See individual bit resets. | 32-bit | Performance Monitors Software Increment Register         |
| 0xCC0  | PMOVSET_ELO [31:0]            | See individual bit resets. | 32-bit | Performance Monitors Overflow Flag Status Set Register   |
| 0xE00  | <a href="#">PMCFGR [31:0]</a> | See individual bit resets. | 32-bit | Performance Monitors Configuration Register              |

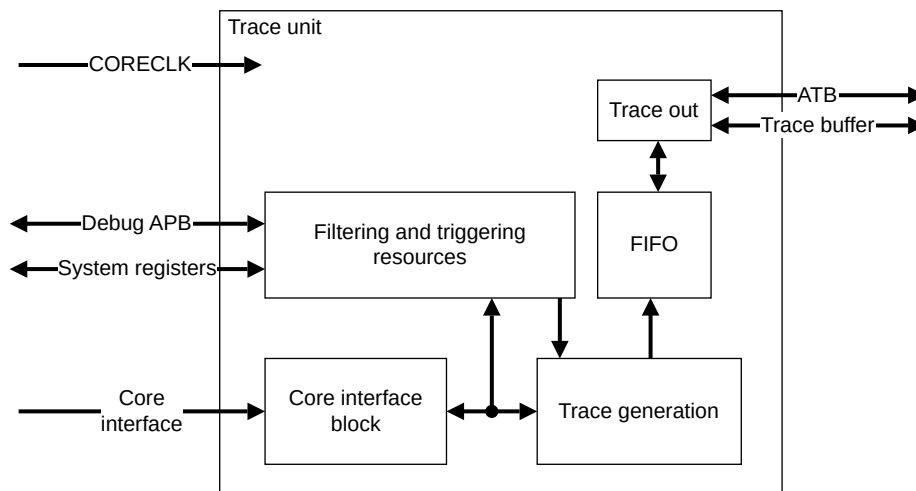
| Offset | Name         | Reset                      | Width  | Description                                                 |
|--------|--------------|----------------------------|--------|-------------------------------------------------------------|
| 0xE04  | PMCR_ELO     | See individual bit resets. | 32-bit | Performance Monitors Control Register                       |
| 0xE20  | PMCEID0      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 0 |
| 0xE24  | PMCEID1      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 1 |
| 0xE28  | PMCEID2      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 2 |
| 0xE2C  | PMCEID3      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 3 |
| 0xE30  | PMSSCR       | See individual bit resets. | 32-bit | PMU Snapshot Capture Register                               |
| 0xE40  | PMMIR [31:0] | See individual bit resets. | 32-bit | Performance Monitors Machine Identification Register        |
| 0xFA8  | PMDEVAFF0    | See individual bit resets. | 32-bit | Performance Monitors Device Affinity register 0             |
| 0xFAC  | PMDEVAFF1    | See individual bit resets. | 32-bit | Performance Monitors Device Affinity register 1             |
| 0xFB0  | PMLAR        | See individual bit resets. | 32-bit | Performance Monitors Lock Access Register                   |
| 0xFB4  | PMLSR        | See individual bit resets. | 32-bit | Performance Monitors Lock Status Register                   |
| 0xFB8  | PMAUTHSTATUS | See individual bit resets. | 32-bit | Performance Monitors Authentication Status register         |
| 0xFBC  | PMDEVARCH    | See individual bit resets. | 32-bit | Performance Monitors Device Architecture register           |
| 0xFC8  | PMDEVID      | See individual bit resets. | 32-bit | Performance Monitors Device ID register                     |
| 0xFCC  | PMDEVTYPE    | See individual bit resets. | 32-bit | Performance Monitors Device Type register                   |
| 0xFD0  | PMPIDR4      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 4   |
| 0xFE0  | PMPIDR0      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 0   |
| 0xFE4  | PMPIDR1      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 1   |
| 0xFE8  | PMPIDR2      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 2   |
| 0xFEC  | PMPIDR3      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 3   |
| 0xFF0  | PMCIDR0      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 0    |
| 0xFF4  | PMCIDR1      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 1    |
| 0xFF8  | PMCIDR2      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 2    |
| 0xFFC  | PMCIDR3      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 3    |

## 18. Embedded Trace Extension support

The Cortex®-X925 core implements the *Embedded Trace Extension* (ETE). The trace unit performs real-time instruction flow tracing based on the ETE. The trace unit is a CoreSight component and is an integral part of the Arm real-time debug solution.

The following figure shows the main components of the trace unit:

**Figure 18-1: Trace unit components**



### Core interface

The core interface monitors and generates P0 elements that are essentially executed branches and exceptions traced in program order.

### Trace generation

The trace generation logic generates various trace packets based on P0 elements.

### Filtering and triggering resources

You can limit the amount of trace data that the trace unit generates by filtering. For example, you can limit trace generation to a certain address range. The trace unit supports other logic analyzer style filtering options. The trace unit can also generate a trigger that is a signal to the Trace Capture Device to stop capturing trace.

### FIFO

The trace unit generates trace in a highly compressed form. The *First In First Out* (FIFO) enables trace bursts to be flattened out. When the FIFO is full, the FIFO signals an overflow. The trace generation logic does not generate any new trace until the FIFO is emptied. This behavior causes a gap in the trace when viewed in the debugger.

## Trace out

Trace from the FIFO is output on the AMBA ATB interface or to the trace buffer.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

## 18.1 Trace unit resources

Trace resources include counters, external input and output signals, and comparators.

The following table shows the trace unit resources, and indicates which of these resources the Cortex®-X925 core implements.

**Table 18-1: Trace unit resources implemented**

| Description                                            | Configuration   |
|--------------------------------------------------------|-----------------|
| Number of resource selection pairs implemented         | 8               |
| Number of external input selectors implemented         | 4               |
| Number of <i>Embedded Trace Extension</i> (ETE) events | 4               |
| Number of counters implemented                         | 2               |
| Reduced function counter implemented                   | Not implemented |
| Number of sequencer states implemented                 | 4               |
| Number of Virtual Machine ID comparators implemented   | 1               |
| Number of Context ID comparators implemented           | 1               |
| Number of address comparator pairs implemented         | 4               |
| Number of single-shot comparator controls              | 1               |
| Number of core comparator inputs implemented           | 0               |
| Data address comparisons implemented                   | Not implemented |
| Number of data value comparators implemented           | 0               |

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

## 18.2 Trace unit generation options

The Cortex®-X925 core trace unit implements a set of generation options.

The following table shows the trace generation options that are implemented in the Cortex®-X925 core trace unit.

**Table 18-2: Trace unit generation options implemented**

| Description                       | Configuration |
|-----------------------------------|---------------|
| Instruction address size in bytes | 8             |

| Description                                                                             | Configuration                                                                        |
|-----------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Data address size in bytes                                                              | 0, because the <i>Embedded Trace Extension</i> (ETE) does not implement data tracing |
| Data value size in bytes                                                                | 0, because the ETE does not implement data tracing                                   |
| Virtual Machine ID size in bytes                                                        | 4                                                                                    |
| Context ID size in bytes                                                                | 4                                                                                    |
| Support for conditional instruction tracing                                             | Not implemented                                                                      |
| Support for tracing of data                                                             | Not implemented                                                                      |
| Support for tracing of load and store instructions because PO elements                  | Not implemented                                                                      |
| Support for cycle counting in the instruction trace                                     | Implemented                                                                          |
| Support for branch broadcast tracing                                                    | Implemented                                                                          |
| Number of events that are supported in the trace                                        | 4                                                                                    |
| Return stack support                                                                    | Implemented                                                                          |
| Tracing of SError exception support                                                     | Implemented                                                                          |
| Instruction trace cycle counting minimum threshold                                      | 4                                                                                    |
| Size of Trace ID                                                                        | 7 bits                                                                               |
| Synchronization period support                                                          | Read/write                                                                           |
| Global timestamp size                                                                   | 64 bits                                                                              |
| Number of cores available for tracing                                                   | 1                                                                                    |
| ATB trigger support                                                                     | Implemented                                                                          |
| Low-power behavior override                                                             | Not implemented                                                                      |
| Stall control support                                                                   | Not implemented                                                                      |
| Support for overflow avoidance                                                          | Not implemented                                                                      |
| Support for using CONTEXTIDR_EL2 in <i>Virtual Machine Identifier</i> (VMID) comparator | Implemented                                                                          |

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

## 18.3 Reset the trace unit

The reset for the trace buffer is the same as a Cold reset for the core. When using the *TRace Buffer Extension* (TRBE), a Warm reset disables the trace buffer and therefore it is not possible to use the trace buffer to capture trace for a Warm reset.

If the trace unit is reset, then tracing stops until the trace unit is reprogrammed and re-enabled. However, if the core is reset using Warm reset, the last few instructions provided by the core before the reset might not be traced.



## 18.4 Program and read the trace unit registers

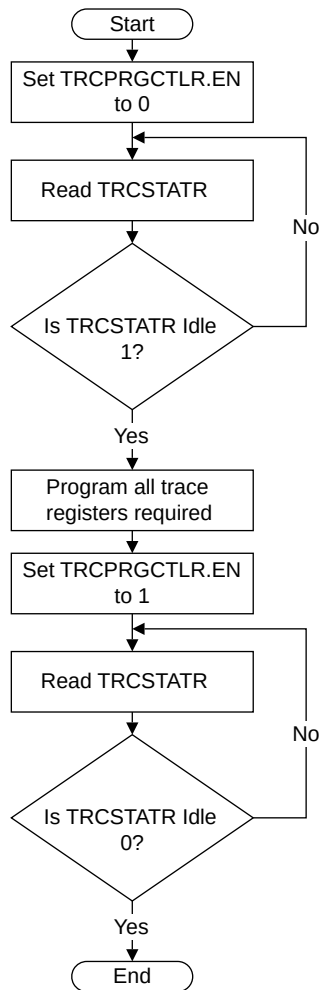
You must program and read the trace unit registers using either the Debug *Advanced Peripheral Bus* (APB) interface or the System register interface.

The core does not have to be in debug state when you program the trace unit registers. When you program the trace unit registers, you must enable all the changes at the same time. Otherwise, if you program the counter, it might start to count based on incorrect events before the correct setup is in place for the trigger condition. To disable the trace unit, use the TRCPRGCTLR.EN bit.

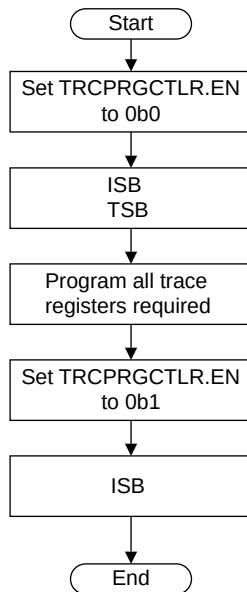
See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about the following trace unit registers:

- Programming Control Register, TRCPRGCTLR
- Trace Status Register, TRCSTATR

The following figure shows the flow for programming trace unit registers using the Debug APB interface:

**Figure 18-2: Programming trace unit registers using the Debug APB interface**

The following figure shows the flow for programming trace unit registers using the System register interface:

**Figure 18-3: Programming trace registers using the System register interface**

## 18.5 Trace unit register interfaces

The Cortex®-X925 core supports an *Advanced Peripheral Bus* (APB) memory-mapped interface and a system register interface to trace unit registers.

Register accesses differ depending on the trace unit state. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for information on the behaviors and access mechanisms.

## 18.6 Interaction with the Performance Monitoring Unit and Debug

The trace unit interacts with the *Performance Monitoring Unit* (PMU) and it can access the PMU events.

### Interaction with the PMU

The Cortex®-X925 core includes a PMU that enables events, such as cache misses and executed instructions, to be counted over time.

The PMU and trace unit function together.

### Use of PMU events by the trace unit

The PMU architectural events are available to the trace unit through the extended input facility. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information about PMU events.

The trace unit uses four extended external input selectors to access the PMU events. Each selector can independently select one of the PMU events, that are then active for the cycles where the relevant events occur. These selected events can then be accessed by any of the event registers within the trace unit.

### Related information

[17. Performance Monitors Extension support](#) on page 119

[17.1 Performance monitors events](#) on page 119

## 18.7 Embedded Trace Extension events

The Cortex®-X925 core trace unit collects events from other units in the design and uses numbers to reference these events.

Other than the events mentioned in [17.1 Performance monitors events](#) on page 119, the events listed in the following table are also exported.

**Table 18-3: ETE events**

| Event number | Event mnemonic | Description                                      |
|--------------|----------------|--------------------------------------------------|
| 0x400D       | PMU_OVFS       | PMU overflow, counters accessible to EL1 and EL0 |
| 0x400E       | TRB_TRIG       | Trace buffer Trigger Event                       |
| 0x400F       | PMU_HOVS       | PMU overflow, counters reserved for use by EL2   |

## 18.8 AArch64 Trace unit registers

The following summary table provides an overview of all Trace unit registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 18-4: Trace unit registers summary**

| Name         | Op0 | Op1 | CRn | CRm      | Op2 | Reset                      | Width  | Description                    |
|--------------|-----|-----|-----|----------|-----|----------------------------|--------|--------------------------------|
| TRCIMSPEC_n_ | 2   | 1   | C0  | 0:m[2:0] | 7   | See individual bit resets. | 64-bit | IMP DEF Register <n>           |
| TRCTRACEIDR  | 2   | 1   | C0  | C0       | 1   | See individual bit resets. | 64-bit | Trace ID Register              |
| TRCVICTLR    | 2   | 1   | C0  | C0       | 2   | See individual bit resets. | 64-bit | ViewInst Main Control Register |

| Name         | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                     |
|--------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------|
| TRCSEQEVR0   | 2   | 1   | C0  | C0  | 4   | See individual bit resets. | 64-bit | Sequencer State Transition Control Register <n> |
| TRCCNTRLDVR0 | 2   | 1   | C0  | C0  | 5   | See individual bit resets. | 64-bit | Counter Reload Value Register <n>               |
| TRCIDR8      | 2   | 1   | C0  | C0  | 6   | See individual bit resets. | 64-bit | ID Register 8                                   |
| TRCIMSPEC0   | 2   | 1   | C0  | C0  | 7   | See individual bit resets. | 64-bit | IMP DEF Register 0                              |
| TRCPRGCTLR   | 2   | 1   | C0  | C1  | 0   | See individual bit resets. | 64-bit | Programming Control Register                    |
| TRCVIIECTLR  | 2   | 1   | C0  | C1  | 2   | See individual bit resets. | 64-bit | ViewInst Include/Exclude Control Register       |
| TRCSEQEVR1   | 2   | 1   | C0  | C1  | 4   | See individual bit resets. | 64-bit | Sequencer State Transition Control Register <n> |
| TRCCNTRLDVR1 | 2   | 1   | C0  | C1  | 5   | See individual bit resets. | 64-bit | Counter Reload Value Register <n>               |
| TRCIDR9      | 2   | 1   | C0  | C1  | 6   | See individual bit resets. | 64-bit | ID Register 9                                   |
| TRCVISSCTLR  | 2   | 1   | C0  | C2  | 2   | See individual bit resets. | 64-bit | ViewInst Start/Stop Control Register            |
| TRCSEQEVR2   | 2   | 1   | C0  | C2  | 4   | See individual bit resets. | 64-bit | Sequencer State Transition Control Register <n> |
| TRCIDR10     | 2   | 1   | C0  | C2  | 6   | See individual bit resets. | 64-bit | ID Register 10                                  |
| TRCSTATR     | 2   | 1   | C0  | C3  | 0   | See individual bit resets. | 64-bit | Trace Status Register                           |
| TRCIDR11     | 2   | 1   | C0  | C3  | 6   | See individual bit resets. | 64-bit | ID Register 11                                  |
| TRCCONFIGR   | 2   | 1   | C0  | C4  | 0   | See individual bit resets. | 64-bit | Trace Configuration Register                    |
| TRCCNTCTLR0  | 2   | 1   | C0  | C4  | 5   | See individual bit resets. | 64-bit | Counter Control Register <n>                    |
| TRCIDR12     | 2   | 1   | C0  | C4  | 6   | See individual bit resets. | 64-bit | ID Register 12                                  |
| TRCCNTCTLR1  | 2   | 1   | C0  | C5  | 5   | See individual bit resets. | 64-bit | Counter Control Register <n>                    |
| TRCIDR13     | 2   | 1   | C0  | C5  | 6   | See individual bit resets. | 64-bit | ID Register 13                                  |
| TRCAUXCTLR   | 2   | 1   | C0  | C6  | 0   | See individual bit resets. | 64-bit | Auxiliary Control Register                      |
| TRCSEQRSTEV  | 2   | 1   | C0  | C6  | 4   | See individual bit resets. | 64-bit | Sequencer Reset Control Register                |
| TRCSEQSTR    | 2   | 1   | C0  | C7  | 4   | See individual bit resets. | 64-bit | Sequencer State Register                        |
| TRCEVENTCTLR | 2   | 1   | C0  | C8  | 0   | See individual bit resets. | 64-bit | Event Control 0 Register                        |

| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                 |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|---------------------------------------------|
| TRCEXTINSELRO | 2   | 1   | C0  | C8  | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCCNTVR0     | 2   | 1   | C0  | C8  | 5   | See individual bit resets. | 64-bit | Counter Value Register <n>                  |
| TRCIDR0       | 2   | 1   | C0  | C8  | 7   | See individual bit resets. | 64-bit | ID Register 0                               |
| TRCEVENTCTL1R | 2   | 1   | C0  | C9  | 0   | See individual bit resets. | 64-bit | Event Control 1 Register                    |
| TRCEXTINSELR1 | 2   | 1   | C0  | C9  | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCCNTVR1     | 2   | 1   | C0  | C9  | 5   | See individual bit resets. | 64-bit | Counter Value Register <n>                  |
| TRCIDR1       | 2   | 1   | C0  | C9  | 7   | See individual bit resets. | 64-bit | ID Register 1                               |
| TRCRSR        | 2   | 1   | C0  | C10 | 0   | See individual bit resets. | 64-bit | Resources Status Register                   |
| TRCEXTINSELR2 | 2   | 1   | C0  | C10 | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCIDR2       | 2   | 1   | C0  | C10 | 7   | See individual bit resets. | 64-bit | ID Register 2                               |
| TRCEXTINSELR3 | 2   | 1   | C0  | C11 | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCIDR3       | 2   | 1   | C0  | C11 | 7   | See individual bit resets. | 64-bit | ID Register 3                               |
| TRCTSCTLR     | 2   | 1   | C0  | C12 | 0   | See individual bit resets. | 64-bit | Timestamp Control Register                  |
| TRCIDR4       | 2   | 1   | C0  | C12 | 7   | See individual bit resets. | 64-bit | ID Register 4                               |
| TRCSYNCPR     | 2   | 1   | C0  | C13 | 0   | See individual bit resets. | 64-bit | Synchronization Period Register             |
| TRCIDR5       | 2   | 1   | C0  | C13 | 7   | See individual bit resets. | 64-bit | ID Register 5                               |
| TRCCCCTLR     | 2   | 1   | C0  | C14 | 0   | See individual bit resets. | 64-bit | Cycle Count Control Register                |
| TRCIDR6       | 2   | 1   | C0  | C14 | 7   | See individual bit resets. | 64-bit | ID Register 6                               |
| TRCBBCTLR     | 2   | 1   | C0  | C15 | 0   | See individual bit resets. | 64-bit | Branch Broadcast Control Register           |
| TRCIDR7       | 2   | 1   | C0  | C15 | 7   | See individual bit resets. | 64-bit | ID Register 7                               |
| TRCSSCCRO     | 2   | 1   | C1  | C0  | 2   | See individual bit resets. | 64-bit | Single-shot Comparator Control Register <n> |
| TRCOSLSR      | 2   | 1   | C1  | C1  | 4   | See individual bit resets. | 64-bit | Trace OS Lock Status Register               |
| TRCRSCTLR2    | 2   | 1   | C1  | C2  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>     |

| Name        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                        |
|-------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------|
| TRCRSCTLR3  | 2   | 1   | C1  | C3  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR4  | 2   | 1   | C1  | C4  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR5  | 2   | 1   | C1  | C5  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR6  | 2   | 1   | C1  | C6  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR7  | 2   | 1   | C1  | C7  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR8  | 2   | 1   | C1  | C8  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCSSCSR0   | 2   | 1   | C1  | C8  | 2   | See individual bit resets. | 64-bit | Single-shot Comparator Control Status Register <n> |
| TRCRSCTLR9  | 2   | 1   | C1  | C9  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR10 | 2   | 1   | C1  | C10 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR11 | 2   | 1   | C1  | C11 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR12 | 2   | 1   | C1  | C12 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR13 | 2   | 1   | C1  | C13 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR14 | 2   | 1   | C1  | C14 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR15 | 2   | 1   | C1  | C15 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCACVR0    | 2   | 1   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR0   | 2   | 1   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR1    | 2   | 1   | C2  | C2  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR1   | 2   | 1   | C2  | C2  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR2    | 2   | 1   | C2  | C4  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR2   | 2   | 1   | C2  | C4  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR3    | 2   | 1   | C2  | C6  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR3   | 2   | 1   | C2  | C6  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR4    | 2   | 1   | C2  | C8  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |

| Name                       | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                              |
|----------------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------|
| TRCACATR4                  | 2   | 1   | C2  | C8  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCACVR5                   | 2   | 1   | C2  | C10 | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR5                  | 2   | 1   | C2  | C10 | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCACVR6                   | 2   | 1   | C2  | C12 | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR6                  | 2   | 1   | C2  | C12 | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCACVR7                   | 2   | 1   | C2  | C14 | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR7                  | 2   | 1   | C2  | C14 | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| <a href="#">TRCCIDCVR0</a> | 2   | 1   | C3  | C0  | 0   | See individual bit resets. | 64-bit | Context Identifier Comparator Value Registers <n>        |
| TRCVMIDCVR0                | 2   | 1   | C3  | C0  | 1   | See individual bit resets. | 64-bit | Virtual Context Identifier Comparator Value Register <n> |
| TRCCIDCCTLRO               | 2   | 1   | C3  | C0  | 2   | See individual bit resets. | 64-bit | Context Identifier Comparator Control Register 0         |
| TRCVMIDCCTLRO              | 2   | 1   | C3  | C2  | 2   | See individual bit resets. | 64-bit | Virtual Context Identifier Comparator Control Register 0 |
| TRCDEVID                   | 2   | 1   | C7  | C2  | 7   | See individual bit resets. | 64-bit | Device Configuration Register                            |
| TRCCLAIMSET                | 2   | 1   | C7  | C8  | 6   | See individual bit resets. | 64-bit | Claim Tag Set Register                                   |
| TRCCLAIMCLR                | 2   | 1   | C7  | C9  | 6   | See individual bit resets. | 64-bit | Claim Tag Clear Register                                 |
| TRCAUTHSTATUS              | 2   | 1   | C7  | C14 | 6   | See individual bit resets. | 64-bit | Authentication Status Register                           |
| TRCDEVARCH                 | 2   | 1   | C7  | C15 | 6   | See individual bit resets. | 64-bit | Device Architecture Register                             |

## 18.9 External ETE registers

The following summary table provides an overview of all memory-mapped ETE registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).



**Table 18-5: ETE registers summary**

| Offset          | Name         | Reset                      | Width  | Description                                        |
|-----------------|--------------|----------------------------|--------|----------------------------------------------------|
| 0x018           | TRCAUXCTLR   | See individual bit resets. | 32-bit | Auxiliary Control Register                         |
| 0x180           | TRCIDR8      | See individual bit resets. | 32-bit | ID Register 8                                      |
| 0x184           | TRCIDR9      | See individual bit resets. | 32-bit | ID Register 9                                      |
| 0x188           | TRCIDR10     | See individual bit resets. | 32-bit | ID Register 10                                     |
| 0x18C           | TRCIDR11     | See individual bit resets. | 32-bit | ID Register 11                                     |
| 0x190           | TRCIDR12     | See individual bit resets. | 32-bit | ID Register 12                                     |
| 0x194           | TRCIDR13     | See individual bit resets. | 32-bit | ID Register 13                                     |
| 0x1C0           | TRCIMSPEC0   | See individual bit resets. | 32-bit | IMP DEF Register 0                                 |
| 0x1E0           | TRCIDR0      | See individual bit resets. | 32-bit | ID Register 0                                      |
| 0x1E4           | TRCIDR1      | See individual bit resets. | 32-bit | ID Register 1                                      |
| 0x1E8           | TRCIDR2      | See individual bit resets. | 32-bit | ID Register 2                                      |
| 0x1EC           | TRCIDR3      | See individual bit resets. | 32-bit | ID Register 3                                      |
| 0x1F0           | TRCIDR4      | See individual bit resets. | 32-bit | ID Register 4                                      |
| 0x1F4           | TRCIDR5      | See individual bit resets. | 32-bit | ID Register 5                                      |
| 0x1F8           | TRCIDR6      | See individual bit resets. | 32-bit | ID Register 6                                      |
| 0x1FC           | TRCIDR7      | See individual bit resets. | 32-bit | ID Register 7                                      |
| 0x2A0 + (4 * n) | TRCSSCSR_n_  | See individual bit resets. | 32-bit | Single-shot Comparator Control Status Register <n> |
| 0xF00           | TRCITCTRL    | See individual bit resets. | 32-bit | Integration Mode Control Register                  |
| 0xFA0           | TRCCCLAIMSET | See individual bit resets. | 32-bit | Claim Tag Set Register                             |
| 0xFA4           | TRCCCLAIMCLR | See individual bit resets. | 32-bit | Claim Tag Clear Register                           |
| 0xFBC           | TRCDEVARCH   | See individual bit resets. | 32-bit | Device Architecture Register                       |
| 0xFC0           | TRCDEVID2    | See individual bit resets. | 32-bit | Device Configuration Register 2                    |
| 0xFC4           | TRCDEVID1    | See individual bit resets. | 32-bit | Device Configuration Register 1                    |
| 0xFC8           | TRCDEVID     | See individual bit resets. | 32-bit | Device Configuration Register                      |
| 0xFCC           | TRCDEVTYPE   | See individual bit resets. | 32-bit | Device Type Register                               |
| 0xFD0           | TRCPIDR4     | See individual bit resets. | 32-bit | Peripheral Identification Register 4               |
| 0xFD4           | TRCPIDR5     | See individual bit resets. | 32-bit | Peripheral Identification Register 5               |
| 0xFD8           | TRCPIDR6     | See individual bit resets. | 32-bit | Peripheral Identification Register 6               |
| 0xFDC           | TRCPIDR7     | See individual bit resets. | 32-bit | Peripheral Identification Register 7               |
| 0xFE0           | TRCPIDR0     | See individual bit resets. | 32-bit | Peripheral Identification Register 0               |
| 0xFE4           | TRCPIDR1     | See individual bit resets. | 32-bit | Peripheral Identification Register 1               |
| 0xFE8           | TRCPIDR2     | See individual bit resets. | 32-bit | Peripheral Identification Register 2               |
| 0xFEC           | TRCPIDR3     | See individual bit resets. | 32-bit | Peripheral Identification Register 3               |
| 0xFF0           | TRCCIDR0     | See individual bit resets. | 32-bit | Component Identification Register 0                |
| 0xFF4           | TRCCIDR1     | See individual bit resets. | 32-bit | Component Identification Register 1                |
| 0xFF8           | TRCCIDR2     | See individual bit resets. | 32-bit | Component Identification Register 2                |
| 0xFFC           | TRCCIDR3     | See individual bit resets. | 32-bit | Component Identification Register 3                |

# 19. Trace Buffer Extension support

The Cortex®-X925 core implements the *TRace Buffer Extension* (TRBE). The TRBE writes the program flow trace generated by the trace unit directly to memory. The TRBE is programmed through System registers.

When enabled, the TRBE can:

- Accept trace data from the trace unit and write it to L2 memory.
- Discard trace data from the trace unit. In this case, the data is lost.
- Reject trace data from the trace unit. In this case, the trace unit retains data until the TRBE accepts it.

When disabled, the TRBE ignores trace data and the trace unit sends trace data to the AMBA® *Trace Bus* (ATB) interface.

## 19.1 Program and read the trace buffer registers

You can program and read the *TRace Buffer Extension* (TRBE) registers using the System register interface.

The core does not have to be in debug state when you program the TRBE registers. When you program the TRBE registers, you must enable all the changes at the same time. Otherwise, if you program the counter, it might start to count based on incorrect events before the correct setup is in place for the trigger condition. To disable the TRBE, use the TRBLIMITR\_EL1.E bit.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for information on the TRBE register behaviors and access mechanisms.

## 19.2 Trace buffer register interface

The Cortex®-X925 core supports a System register interface to *TRace Buffer Extension* (TRBE) registers.

Register accesses differ depending on the TRBE state. See the [Arm® Architecture Reference Manual for A-profile architecture](#) for information on the behaviors and access mechanisms.

## 20. Activity Monitors Extension support

The Cortex®-X925 core implements the Activity Monitors Extension to the Arm®v8.4-A architecture. Activity monitoring has features similar to performance monitoring features, but it is intended for system management use whereas performance monitoring is aimed at user and debug applications.

The activity monitors provide useful information for system power management and persistent monitoring. The activity monitors are read-only in operation and their configuration is limited to the highest Exception level implemented.

The Cortex®-X925 core implements seven counters in two groups, each of which is a 64-bit counter that counts a fixed event. Group 0 has four counters 0-3, and Group 1 has three counters 10-12.

### 20.1 Activity monitors access

The Cortex®-X925 core supports access to activity monitors from the System register interface and supports read-only memory-mapped access using the utility bus interface.

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for information on the memory mapping of these registers.

#### Access enable bit

There are multiple access enable bits associated with the Activity Monitors System registers:

- AMUSERENR\_ELO.EN controls access from EL0 to the Activity Monitors System registers
- CPTR\_EL2.TAM controls access from EL0 and EL1 to the Activity Monitors System registers
- CPTR\_EL3.TAM controls access from EL0, EL1, and EL2 to the Activity Monitors Extension System registers



AMUSERENR\_ELO.EN is configurable at EL1, EL2, and EL3. All other controls, as well as the value of the counters, are configurable only at the highest implemented Exception level.

---

For a detailed description of access controls for the registers, see the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### System register access

The activity monitors are accessible using the `MRS` and `MSR` instructions.

## External memory-mapped access

Activity monitors can be memory-mapped accessed from the utility bus interface. In this case, the Activity Monitors registers only provide read access to the Activity Monitor Event Counter registers.

The base address for *Activity Monitoring Unit* (AMU) registers on the utility bus interface is  $0x\langle n \rangle 90000$ , where  $n$  is the Cortex®-X925 core instance number in the DSU-120 DynaMIQ™ cluster.

These registers are treated as RAZ/WI if either:

- The register is marked as Reserved.
- The register is accessed in the wrong Security state.

## 20.2 Activity monitors counters

The Cortex®-X925 core implements four activity monitors counters, 0-3, and three auxiliary counters, 10-12 that map to specific *Activity Monitoring Unit* (AMU) events.

Each counter has the following characteristics:

- All events are counted in 64-bit wrapping counters that overflow when they wrap. There is no support for overflow status indication or interrupts.
- Any change in clock frequency can affect any counter. For example, when a *WFI*, *WFE*, *WFIIT*, or *WFET* instruction stops the clock.
- Events 0-3 and auxiliary events 10-12 are fixed, and the *AMEVTYPER0* $\langle n \rangle$ \_ELO and *AMEVTYPER1* $\langle n \rangle$ \_ELO evtCount bits are read-only.
- The activity monitor counters are reset to zero on a Cold reset of the power domain of the core. When the core is not in reset, activity monitoring is available.

## 20.3 Activity monitors events

Activity monitors events in the Cortex®-X925 core are either fixed or programmable, and they map to the activity monitors counters.

The following table shows the mapping of counters to fixed events.

**Table 20-1: Mapping of counters to fixed events**

| Activity monitor counter $\langle n \rangle$ | Event      | Event number | Description               |
|----------------------------------------------|------------|--------------|---------------------------|
| AMEVCNTR00                                   | CPU_CYCLES | 0x0011       | Core frequency cycles     |
| AMEVCNTR01                                   | CNT_CYCLES | 0x4004       | Constant frequency cycles |

| Activity monitor counter <n> | Event                | Event number | Description                                                                                                                                                                                                                                     |
|------------------------------|----------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AMEVCNTR02                   | INSTR_RETIRED        | 0x0008       | Instruction architecturally executed<br><br>This counter increments for every instruction that is executed architecturally, including instructions that fail their condition code check.                                                        |
| AMEVCNTR03                   | STALL_BACKEND_MEM    | 0x4005       | Memory stall cycles<br><br>This counter counts cycles in which the core is unable to dispatch instructions from the front end to the back end due to a back end stall caused by a miss in the last level of cache within the core clock domain. |
| AMEVCNTR10                   | MPMM_THRESHOLD_GEAR0 | 0x0300       | <i>Maximum Power Mitigation System (MPMM) Gear 0 activity period threshold exceeded</i>                                                                                                                                                         |
| AMEVCNTR11                   | MPMM_THRESHOLD_GEAR1 | 0x0301       | <i>Maximum Power Mitigation System (MPMM) Gear 1 activity period threshold exceeded</i>                                                                                                                                                         |
| AMEVCNTR12                   | MPMM_THRESHOLD_GEAR2 | 0x0302       | <i>Maximum Power Mitigation System (MPMM) Gear 2 activity period threshold exceeded</i>                                                                                                                                                         |

### Related information

[4.5.1 Maximum Power Mitigation Mechanism](#) on page 46

## 20.4 AArch64 Activity Monitors registers summary

The following summary table provides an overview of all Activity Monitors registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 20-2: Activity Monitors registers summary**

| Name                       | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                            |
|----------------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------------|
| AMCR_ELO                   | 3   | 3   | C13 | C2  | 0   | See individual bit resets. | 64-bit | Activity Monitors Control Register                     |
| <a href="#">AMCFGR_ELO</a> | 3   | 3   | C13 | C2  | 1   | See individual bit resets. | 64-bit | Activity Monitors Configuration Register               |
| <a href="#">AMCGCR_ELO</a> | 3   | 3   | C13 | C2  | 2   | See individual bit resets. | 64-bit | Activity Monitors Counter Group Configuration Register |
| AMUSERENR_ELO              | 3   | 3   | C13 | C2  | 3   | See individual bit resets. | 64-bit | Activity Monitors User Enable Register                 |
| AMCNTENCLR0_ELO            | 3   | 3   | C13 | C2  | 4   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Clear Register 0        |

| Name                            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                     |
|---------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------|
| AMCNTENSET0_ELO                 | 3   | 3   | C13 | C2  | 5   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Set Register 0   |
| AMCNTENCLR1_ELO                 | 3   | 3   | C13 | C3  | 0   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Clear Register 1 |
| AMCNTENSET1_ELO                 | 3   | 3   | C13 | C3  | 1   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Set Register 1   |
| <a href="#">AMEVCNTR00_ELO</a>  | 3   | 3   | C13 | C4  | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0     |
| <a href="#">AMEVCNTR01_ELO</a>  | 3   | 3   | C13 | C4  | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0     |
| <a href="#">AMEVCNTR02_ELO</a>  | 3   | 3   | C13 | C4  | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0     |
| <a href="#">AMEVCNTR03_ELO</a>  | 3   | 3   | C13 | C4  | 3   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0     |
| <a href="#">AMEVTYPER00_ELO</a> | 3   | 3   | C13 | C6  | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0        |
| <a href="#">AMEVTYPER01_ELO</a> | 3   | 3   | C13 | C6  | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0        |
| <a href="#">AMEVTYPER02_ELO</a> | 3   | 3   | C13 | C6  | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0        |
| <a href="#">AMEVTYPER03_ELO</a> | 3   | 3   | C13 | C6  | 3   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0        |
| <a href="#">AMEVCNTR10_ELO</a>  | 3   | 3   | C13 | C12 | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 1     |
| <a href="#">AMEVCNTR11_ELO</a>  | 3   | 3   | C13 | C12 | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 1     |
| <a href="#">AMEVCNTR12_ELO</a>  | 3   | 3   | C13 | C12 | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 1     |
| <a href="#">AMEVTYPER10_ELO</a> | 3   | 3   | C13 | C14 | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 1        |
| <a href="#">AMEVTYPER11_ELO</a> | 3   | 3   | C13 | C14 | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 1        |
| <a href="#">AMEVTYPER12_ELO</a> | 3   | 3   | C13 | C14 | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 1        |

## 20.5 External AMU registers

The following summary table provides an overview of all memory-mapped AMU registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 20-3: AMU registers summary**

| Offset | Name                               | Reset                      | Width  | Description                                              |
|--------|------------------------------------|----------------------------|--------|----------------------------------------------------------|
| 0x0    | <a href="#">AMEVCNTR00 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x4    | <a href="#">AMEVCNTR00 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x8    | <a href="#">AMEVCNTR01 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0xC    | <a href="#">AMEVCNTR01 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x10   | <a href="#">AMEVCNTR02 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x14   | <a href="#">AMEVCNTR02 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x18   | <a href="#">AMEVCNTR03 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x1C   | <a href="#">AMEVCNTR03 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0              |
| 0x100  | <a href="#">AMEVCNTR10 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1              |
| 0x104  | <a href="#">AMEVCNTR10 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1              |
| 0x108  | <a href="#">AMEVCNTR11 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1              |
| 0x10C  | <a href="#">AMEVCNTR11 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1              |
| 0x110  | <a href="#">AMEVCNTR12 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1              |
| 0x114  | <a href="#">AMEVCNTR12 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1              |
| 0x400  | <a href="#">AMEVTYPER00</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x404  | <a href="#">AMEVTYPER01</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x408  | <a href="#">AMEVTYPER02</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x40C  | <a href="#">AMEVTYPER03</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x480  | <a href="#">AMEVTYPER10</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 1                 |
| 0x484  | <a href="#">AMEVTYPER11</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 1                 |
| 0x488  | <a href="#">AMEVTYPER12</a>        | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 1                 |
| 0xC00  | <a href="#">AMCNTENSET0</a>        | See individual bit resets. | 32-bit | Activity Monitors Count Enable Set Register 0            |
| 0xC04  | <a href="#">AMCNTENSET1</a>        | See individual bit resets. | 32-bit | Activity Monitors Count Enable Set Register 1            |
| 0xC20  | <a href="#">AMCNTENCLR0</a>        | See individual bit resets. | 32-bit | Activity Monitors Count Enable Clear Register 0          |
| 0xC24  | <a href="#">AMCNTENCLR1</a>        | See individual bit resets. | 32-bit | Activity Monitors Count Enable Clear Register 1          |
| 0xCE0  | <a href="#">AMCGCR</a>             | See individual bit resets. | 32-bit | Activity Monitors Counter Group Configuration Register   |
| 0xE00  | <a href="#">AMCFGR</a>             | See individual bit resets. | 32-bit | Activity Monitors Configuration Register                 |
| 0xE04  | <a href="#">AMCR</a>               | See individual bit resets. | 32-bit | Activity Monitors Control Register                       |
| 0xE08  | <a href="#">AMIIDR</a>             | See individual bit resets. | 32-bit | Activity Monitors Implementation Identification Register |
| 0xFA8  | <a href="#">AMDEVAFF0</a>          | See individual bit resets. | 32-bit | Activity Monitors Device Affinity Register 0             |
| 0xFAC  | <a href="#">AMDEVAFF1</a>          | See individual bit resets. | 32-bit | Activity Monitors Device Affinity Register 1             |
| 0xFBC  | <a href="#">AMDEVARCH</a>          | See individual bit resets. | 32-bit | Activity Monitors Device Architecture Register           |
| 0xFCC  | <a href="#">AMDEVTYPE</a>          | See individual bit resets. | 32-bit | Activity Monitors Device Type Register                   |
| 0xFD0  | <a href="#">AMPIDR4</a>            | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 4   |
| 0xFE0  | <a href="#">AMPIDR0</a>            | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 0   |
| 0xFE4  | <a href="#">AMPIDR1</a>            | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 1   |
| 0xFE8  | <a href="#">AMPIDR2</a>            | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 2   |

| Offset | Name                    | Reset                      | Width  | Description                                            |
|--------|-------------------------|----------------------------|--------|--------------------------------------------------------|
| 0xFEC  | <a href="#">AMPIDR3</a> | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 3 |
| 0xFF0  | <a href="#">AMCIDR0</a> | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 0  |
| 0xFF4  | <a href="#">AMCIDR1</a> | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 1  |
| 0xFF8  | <a href="#">AMCIDR2</a> | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 2  |
| 0xFFC  | <a href="#">AMCIDR3</a> | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 3  |



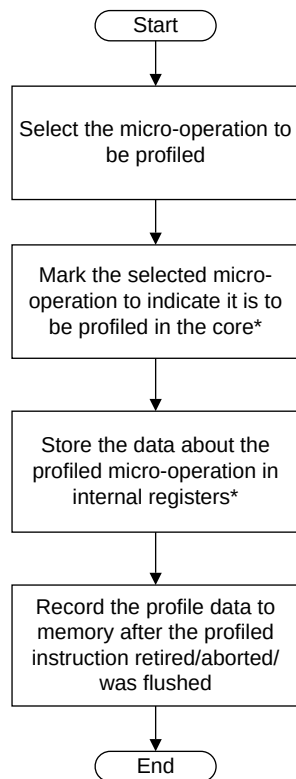
## 21. Statistical Profiling Extension support

The Cortex®-X925 core implements the optional *Statistical Profiling Extension* (SPE) to the Arm®v8.7-A architecture. The SPE provides a statistical view of the performance characteristics of executed instructions that software writers can use to optimize their code for better performance.

The Cortex®-X925 core profiles micro-operations to minimize the amount of logic necessary to support the SPE.

The following figure shows the SPE behavior in the Cortex®-X925 core.

**Figure 21-1: SPE behavior**



\* Throughout the lifetime of the micro-operation in the core

Profiles are collected periodically and a down-counter drives the selection of the micro-operations to be profiled. This counter counts the number of speculative micro-operations that are dispatched, decremented once for each micro-operation. When the counter reaches zero, a micro-operation is identified as being sampled and is profiled throughout its lifetime in the core.

SPE profiles are written to memory using a *Virtual Address* (VA), which means that writes of profiles must have access to the *Memory Management Unit* (MMU) to translate a VA to a *Physical Address* (PA), and must have a means to be written to memory.



Profiling is expected to be largely non-intrusive to the performance of the core. The performance of the core is not meaningfully perturbed while profiling is taking place. The rate of occurrence depends on the sampling rate. You can specify a sampling rate that is meaningfully intrusive to the performance of the core. Arm recommends that the minimum sampling interval is once per 1024 micro-operations. This value is communicated to software through PMSIDR\_EL1.Interval, bits[11:8].

See the [Arm® Architecture Reference Manual for A-profile architecture](#) for more information.

## 21.1 Statistical Profiling Extension events packet

The events packet indicates the **IMPLEMENTATION DEFINED** events that the sampled operation generated.

The following table shows the events defined in the 32-bit events packet implemented in the Cortex®-X925 core.

**Table 21-1: SPE events packet**

| Bits    | Definition                                              |
|---------|---------------------------------------------------------|
| [31:19] | Reserved                                                |
| [18]    | Empty predicate                                         |
| [17]    | Partial predicate                                       |
| [16:13] | Reserved                                                |
| [12]    | Late prefetch                                           |
| [11]    | Data alignment flag                                     |
| [10]    | Remote access                                           |
| [9]     | Last level cache miss                                   |
| [8]     | Last level cache access                                 |
| [7]     | Branch mispredicted                                     |
| [6]     | Not taken                                               |
| [5]     | L1 data cache <i>Translation Lookaside Buffer</i> (TLB) |
| [4]     | TLB access                                              |
| [3]     | L1 data cache refill                                    |
| [2]     | L1 data cache access                                    |
| [1]     | Architecturally retired                                 |
| [0]     | Generated exception                                     |

## 21.2 Statistical Profiling Extension data source packet

The data source packet indicates where the data returned for a load or store operation was sourced.

The following table shows the data source defined in the 8-bit data source packet implemented in the Cortex®-X925 core.

**Table 21-2: SPE data source packet**

| Value  | Name                                |
|--------|-------------------------------------|
| 0b0000 | L1 data cache                       |
| 0b1000 | L2 cache                            |
| 0b1001 | Peer core                           |
| 0b1010 | Local cluster                       |
| 0b1011 | System cache                        |
| 0b1100 | Peer cluster                        |
| 0b1101 | Remote                              |
| 0b1110 | Dynamic Random Access Memory (DRAM) |

## 21.3 AArch64 Statistical Profiling Extension registers

The following summary table provides an overview of all Statistical Profiling Extension registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table 21-3: Statistical Profiling Extension registers summary**

| Name                         | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                  |
|------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------|
| PMSCR_EL1                    | 3   | 0   | C9  | C9  | 0   | See individual bit resets. | 64-bit | Statistical Profiling Control Register (EL1) |
| <a href="#">PMSNEVFR_EL1</a> | 3   | 0   | C9  | C9  | 1   | See individual bit resets. | 64-bit | Sampling Inverted Event Filter Register      |
| PMSICR_EL1                   | 3   | 0   | C9  | C9  | 2   | See individual bit resets. | 64-bit | Sampling Interval Counter Register           |
| PMSIRR_EL1                   | 3   | 0   | C9  | C9  | 3   | See individual bit resets. | 64-bit | Sampling Interval Reload Register            |
| PMSFCR_EL1                   | 3   | 0   | C9  | C9  | 4   | See individual bit resets. | 64-bit | Sampling Filter Control Register             |
| <a href="#">PMSEVFR_EL1</a>  | 3   | 0   | C9  | C9  | 5   | See individual bit resets. | 64-bit | Sampling Event Filter Register               |
| PMSLATFR_EL1                 | 3   | 0   | C9  | C9  | 6   | See individual bit resets. | 64-bit | Sampling Latency Filter Register             |
| <a href="#">PMSIDR_EL1</a>   | 3   | 0   | C9  | C9  | 7   | See individual bit resets. | 64-bit | Sampling Profiling ID Register               |

| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                  |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------|
| PMBLIMITR_EL1 | 3   | 0   | C9  | C10 | 0   | See individual bit resets. | 64-bit | Profiling Buffer Limit Address Register      |
| PMBPTR_EL1    | 3   | 0   | C9  | C10 | 1   | See individual bit resets. | 64-bit | Profiling Buffer Write Pointer Register      |
| PMBSR_EL1     | 3   | 0   | C9  | C10 | 3   | See individual bit resets. | 64-bit | Profiling Buffer Status/syndrome Register    |
| PMBIDR_EL1    | 3   | 0   | C9  | C10 | 7   | See individual bit resets. | 64-bit | Profiling Buffer ID Register                 |
| PMSCR_EL2     | 3   | 4   | C9  | C9  | 0   | See individual bit resets. | 64-bit | Statistical Profiling Control Register (EL2) |

# Appendix A AArch64 System registers

This appendix contains the descriptions for the Cortex®-X925 AArch64 registers.

This manual does not provide a complete list of registers. Read this manual together with the [Arm® Architecture Reference Manual for A-profile architecture](#).

## A.1 AArch64 registers summary

The following summary table provides an overview of all registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-1: registers summary**

| Name                              | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                    |
|-----------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------|
| <a href="#">IMP_CPUPPMPCR_EL1</a> | 3   | 0   | C15 | C2  | 4   | See individual bit resets. | 64-bit | Performance and Power Management PDP Control Register          |
| <a href="#">IMP_CPUPPMCR_EL3</a>  | 3   | 6   | C15 | C2  | 0   | See individual bit resets. | 64-bit | Global Performance and Power Management Configuration Register |
| <a href="#">IMP_CPUMPMCR_EL3</a>  | 3   | 6   | C15 | C2  | 1   | See individual bit resets. | 64-bit | Global MPMM Control Register                                   |
| <a href="#">IMP_CPUPPMCR4_EL3</a> | 3   | 6   | C15 | C2  | 4   | See individual bit resets. | 64-bit | CPU Power Performance Management Control Register              |
| <a href="#">IMP_CPUPPMCR5_EL3</a> | 3   | 6   | C15 | C2  | 5   | See individual bit resets. | 64-bit | CPU Power Performance Management Control Register              |
| <a href="#">IMP_CPUPPMCR6_EL3</a> | 3   | 6   | C15 | C2  | 6   | See individual bit resets. | 64-bit | CPU Power Performance Management Control Register              |

### A.1.1 IMP\_CPUPPMPCR\_EL1, Performance and Power Management PDP Control Register

Provides **IMPLEMENTATION DEFINED** control of the Performance Defined Power (PDP) feature.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

#### Functional group

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-1: AArch64\_imp\_cpupmpdpcr\_el1 bit assignments

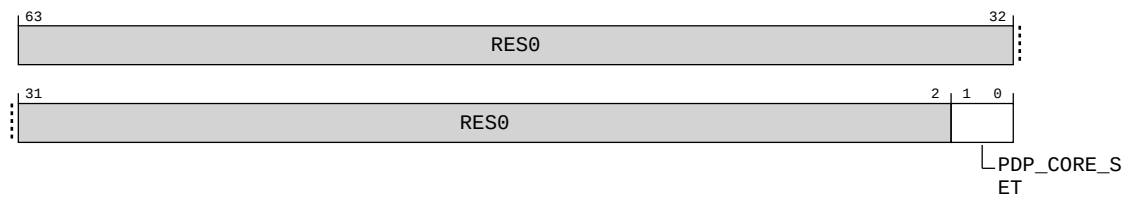


Table A-2: IMP\_CPUPMPDPCR\_EL1 bit descriptions

| Bits   | Name         | Description                                                                                                                                                                                                                        | Reset |
|--------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:2] | RES0         | Reserved                                                                                                                                                                                                                           | RES0  |
| [1:0]  | PDP_CORE_SET | Core PDP Aggressiveness<br><br><b>0b00</b><br>Disable PDP<br><br><b>0b01</b><br>Enable PDP at low aggressiveness<br><br><b>0b10</b><br>Enable PDP at medium aggressiveness<br><br><b>0b11</b><br>Enable PDP at high aggressiveness | 0b00  |

Access

MRS <Xt>, S3\_0\_C15\_C2\_4

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0010 | 0b100 |

MSR S3\_0\_C15\_C2\_4, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0010 | 0b100 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C2\_4

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUPPMPDPCR_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUPPMPDPCR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPPMPDPCR_EL1;

```

MSR S3\_0\_C15\_C2\_4, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        IMP_CPUPPMPDPCR_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    IMP_CPUPPMPDPCR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUPPMPDPCR_EL1 = X[t, 64];

```

## A.1.2 IMP\_CPUPPMCR\_EL3, Global Performance and Power Management Configuration Register

Provides **IMPLEMENTATION DEFINED** control and discovery of the Performance and Power Management (PPM) features.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

#### Access type

See bit descriptions

#### Reset value

```
xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx x011 xxxx x011 xxxx xx00
```



Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-2: AArch64\_imp\_cpuppmcr\_el3 bit assignments**

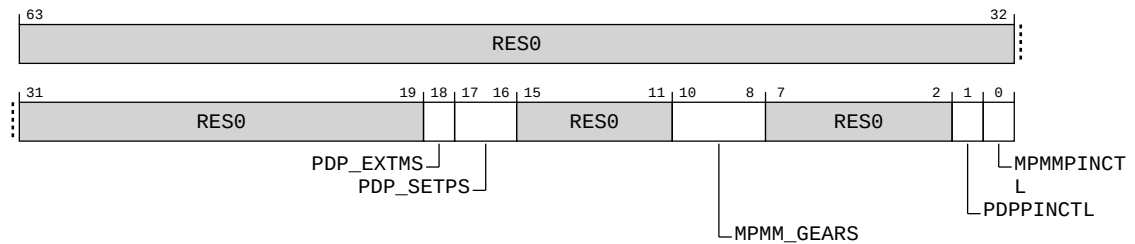


**Table A-5: IMP\_CPUPPMCR\_EL3 bit descriptions**

| Bits    | Name       | Description                                                                                                                       | Reset |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:19] | RES0       | Reserved                                                                                                                          | RES0  |
| [18]    | PDP_EXTMS  | External memory system PDP control<br><b>0b0</b><br>Independent external memory system PDP control is not implemented             | 0b0   |
| [17:16] | PDP_SETPS  | Number of PDP Setpoints Implemented<br><b>0b11</b><br>3 PDP setpoints are implemented                                             | 0b11  |
| [15:11] | RES0       | Reserved                                                                                                                          | RES0  |
| [10:8]  | MPMM_GEARs | Number of MPMM Gears Implemented<br><b>0b011</b><br>3 MPMM gears are implemented                                                  | 0b011 |
| [7:2]   | RES0       | Reserved                                                                                                                          | RES0  |
| [1]     | PDPPINCTL  | PDP Pin Control Enabled<br><b>0b0</b><br>PDP control through SPR and utility bus<br><b>0b1</b><br>PDP control through pin only    | 0b0   |
| [0]     | MPMMPINCTL | MPMM Pin Control Enabled<br><b>0b0</b><br>MPMM control through SPR and utility bus<br><b>0b1</b><br>MPMM control through pin only | 0b0   |



Access

MRS <Xt>, S3\_6\_C15\_C2\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b000 |

MSR S3\_6\_C15\_C2\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b000 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C2\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPPMCR_EL3;
```

MSR S3\_6\_C15\_C2\_0, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    IMP_CPUPPMCR_EL3 = X[t, 64];
```

A.1.3 IMP\_CPUMPMMCR\_EL3, Global MPMM Control Register

Provides **IMPLEMENTATION DEFINED** control of the Maximum Power Mitigation Mechanism (MPMM) feature.

Configurations

This register is available in all configurations.

Attributes

Width

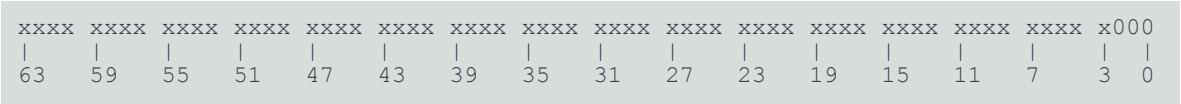
64

Functional group

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-3: AArch64\_imp\_cpumpmmcr\_el3 bit assignments

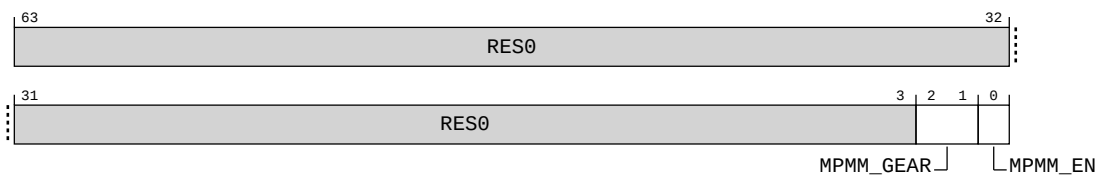


Table A-8: IMP\_CPUMPMPCR\_EL3 bit descriptions

| Bits   | Name      | Description                                                                                                                                                                          | Reset |
|--------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:3] | RES0      | Reserved                                                                                                                                                                             | RES0  |
| [2:1]  | MPMM_GEAR | MPMM Gear Select<br><br><b>0b00</b><br>Select MPMM Gear 0<br><br><b>0b01</b><br>Select MPMM Gear 1<br><br><b>0b10</b><br>Select MPMM Gear 2<br><br><b>0b11</b><br>Select MPMM Gear 3 | 0b00  |
| [0]    | MPMM_EN   | MPMM Master Enable<br><br><b>0b0</b><br>MPMM is disabled<br><br><b>0b1</b><br>MPMM is enabled                                                                                        | 0b0   |

Access

MRS <Xt>, S3\_6\_C15\_C2\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b001 |

MSR S3\_6\_C15\_C2\_1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b001 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C2\_1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUMPMCR_EL3;
```

MSR S3\_6\_C15\_C2\_1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    IMP_CPUMPMCR_EL3 = X[t, 64];
```

A.1.4 IMP\_CPUPPMCR4\_EL3, CPU Power Performance Management Control Register

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-4: AArch64\_imp\_cpuppmcr4\_el3 bit assignments

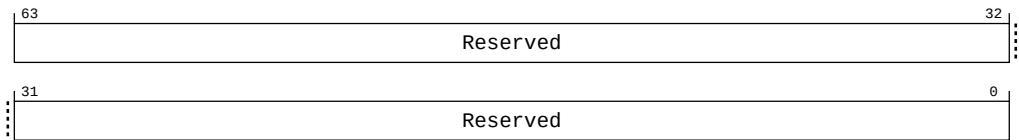


Table A-11: IMP\_CPUPPMCR4\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_6\_C15\_C2\_4

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b100 |

MSR S3\_6\_C15\_C2\_4, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b100 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C2\_4

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPPMCR4_EL3;
```

MSR S3\_6\_C15\_C2\_4, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
```

```
elseif PSTATE.EL == EL3 then
    IMP_CPUPPMCR4_EL3 = X[t, 64];
```

### A.1.5 IMP\_CPUPPMCR5\_EL3, CPU Power Performance Management Control Register

This register contains control bits that affect the CPU behavior

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

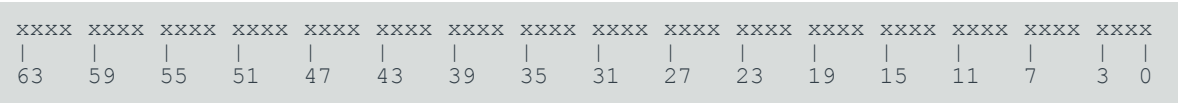
64

##### Functional group

##### Access type

See bit descriptions

##### Reset value



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-5: AArch64\_imp\_cpuppmcr5\_el3 bit assignments

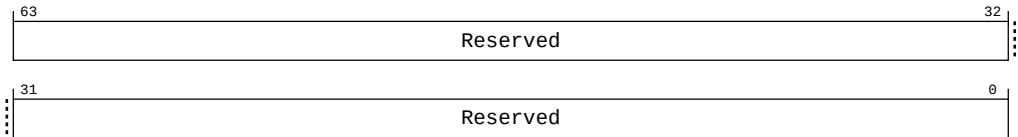


Table A-14: IMP\_CPUPPMCR5\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

#### Access

MRS <Xt>, S3\_6\_C15\_C2\_5

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b101 |

MSR S3\_6\_C15\_C2\_5, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b101 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C2\_5

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPPMCR5_EL3;
```

MSR S3\_6\_C15\_C2\_5, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    IMP_CPUPPMCR5_EL3 = X[t, 64];
```

A.1.6 IMP\_CPUPPMCR6\_EL3, CPU Power Performance Management Control Register

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Access type

See bit descriptions

Reset value

```
xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
```



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-6: AArch64\_imp\_cpuppmcr6\_el3 bit assignments

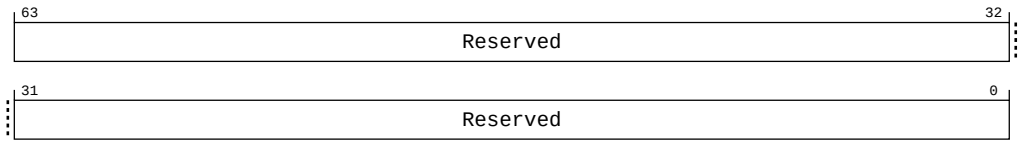


Table A-17: IMP\_CPUPPMCR6\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_6\_C15\_C2\_6

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b110 |

MSR S3\_6\_C15\_C2\_6, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b110 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C2\_6

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPPMCR6_EL3;
```

MSR S3\_6\_C15\_C2\_6, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
```

```

elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    IMP_CPUPPMCR6_EL3 = X[t, 64];

```

## A.2 AArch64 Generic System Control registers summary

The following summary table provides an overview of all Generic System Control registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-20: Generic System Control registers summary**

| Name                      | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                 |
|---------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------------|
| <a href="#">ACTLR_EL1</a> | 3   | 0   | C1  | C0  | 1   | See individual bit resets. | 64-bit | Auxiliary Control Register (EL1)                            |
| RGSR_EL1                  | 3   | 0   | C1  | C0  | 5   | See individual bit resets. | 64-bit | Random Allocation Tag Seed Register.                        |
| GCR_EL1                   | 3   | 0   | C1  | C0  | 6   | See individual bit resets. | 64-bit | Tag Control Register.                                       |
| TTBR0_EL1                 | 3   | 0   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Translation Table Base Register 0 (EL1)                     |
| TTBR1_EL1                 | 3   | 0   | C2  | C0  | 1   | See individual bit resets. | 64-bit | Translation Table Base Register 1 (EL1)                     |
| TCR_EL1                   | 3   | 0   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Translation Control Register (EL1)                          |
| APIAKeyLo_EL1             | 3   | 0   | C2  | C1  | 0   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Instruction (bits[63:0])   |
| APIAKeyHi_EL1             | 3   | 0   | C2  | C1  | 1   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Instruction (bits[127:64]) |
| APIBKeyLo_EL1             | 3   | 0   | C2  | C1  | 2   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Instruction (bits[63:0])   |
| APIBKeyHi_EL1             | 3   | 0   | C2  | C1  | 3   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Instruction (bits[127:64]) |
| APDAKeyLo_EL1             | 3   | 0   | C2  | C2  | 0   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Data (bits[63:0])          |
| APDAKeyHi_EL1             | 3   | 0   | C2  | C2  | 1   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Data (bits[127:64])        |



| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                           |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------|
| APDBKeyLo_EL1 | 3   | 0   | C2  | C2  | 2   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Data (bits[63:0])    |
| APDBKeyHi_EL1 | 3   | 0   | C2  | C2  | 3   | See individual bit resets. | 64-bit | Pointer Authentication Key B for Data (bits[127:64])  |
| APGAKeyLo_EL1 | 3   | 0   | C2  | C3  | 0   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Code (bits[63:0])    |
| APGAKeyHi_EL1 | 3   | 0   | C2  | C3  | 1   | See individual bit resets. | 64-bit | Pointer Authentication Key A for Code (bits[127:64])  |
| SPSel         | 3   | 0   | C4  | C2  | 0   | See individual bit resets. | 64-bit | Stack Pointer Select                                  |
| CurrentEL     | 3   | 0   | C4  | C2  | 2   | See individual bit resets. | 64-bit | Current Exception Level                               |
| PAN           | 3   | 0   | C4  | C2  | 3   | See individual bit resets. | 64-bit | Privileged Access Never                               |
| UAO           | 3   | 0   | C4  | C2  | 4   | See individual bit resets. | 64-bit | User Access Override                                  |
| AFSR0_EL1     | 3   | 0   | C5  | C1  | 0   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 0 (EL1)               |
| AFSR1_EL1     | 3   | 0   | C5  | C1  | 1   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 1 (EL1)               |
| ESR_EL1       | 3   | 0   | C5  | C2  | 0   | See individual bit resets. | 64-bit | Exception Syndrome Register (EL1)                     |
| TFSR_EL1      | 3   | 0   | C5  | C6  | 0   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL1)                       |
| TFSREQ_EL1    | 3   | 0   | C5  | C6  | 1   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL0).                      |
| FAR_EL1       | 3   | 0   | C6  | C0  | 0   | See individual bit resets. | 64-bit | Fault Address Register (EL1)                          |
| PAR_EL1       | 3   | 0   | C7  | C4  | 0   | See individual bit resets. | 64-bit | Physical Address Register                             |
| MAIR_EL1      | 3   | 0   | C10 | C2  | 0   | See individual bit resets. | 64-bit | Memory Attribute Indirection Register (EL1)           |
| AMAIR_EL1     | 3   | 0   | C10 | C3  | 0   | See individual bit resets. | 64-bit | Auxiliary Memory Attribute Indirection Register (EL1) |
| LORSA_EL1     | 3   | 0   | C10 | C4  | 0   | See individual bit resets. | 64-bit | LORegion Start Address (EL1)                          |
| LOREA_EL1     | 3   | 0   | C10 | C4  | 1   | See individual bit resets. | 64-bit | LORegion End Address (EL1)                            |
| LORN_EL1      | 3   | 0   | C10 | C4  | 2   | See individual bit resets. | 64-bit | LORegion Number (EL1)                                 |
| LORC_EL1      | 3   | 0   | C10 | C4  | 3   | See individual bit resets. | 64-bit | LORegion Control (EL1)                                |
| LORID_EL1     | 3   | 0   | C10 | C4  | 7   | See individual bit resets. | 64-bit | LORegionID (EL1)                                      |
| VBAR_EL1      | 3   | 0   | C12 | C0  | 0   | See individual bit resets. | 64-bit | Vector Base Address Register (EL1)                    |

| Name                   | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                      |
|------------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------|
| ISR_EL1                | 3   | 0   | C12 | C1  | 0   | See individual bit resets. | 64-bit | Interrupt Status Register                        |
| CONTEXTIDR_EL1         | 3   | 0   | C13 | C0  | 1   | See individual bit resets. | 64-bit | Context ID Register (EL1)                        |
| TPIDR_EL1              | 3   | 0   | C13 | C0  | 4   | See individual bit resets. | 64-bit | EL1 Software Thread ID Register                  |
| SCXTNUM_EL1            | 3   | 0   | C13 | C0  | 7   | See individual bit resets. | 64-bit | EL1 Read/Write Software Context Number           |
| IMP_CPUACTLR_EL1       | 3   | 0   | C15 | C1  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register (EL1)             |
| IMP_CPUACTLR2_EL1      | 3   | 0   | C15 | C1  | 1   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 2 (EL1)           |
| IMP_CPUACTLR3_EL1      | 3   | 0   | C15 | C1  | 2   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 3 (EL1)           |
| IMP_CPUACTLR4_EL1      | 3   | 0   | C15 | C1  | 3   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 4 (EL1)           |
| IMP_CPUECTLR_EL1       | 3   | 0   | C15 | C1  | 4   | See individual bit resets. | 64-bit | CPU Extended Control Register                    |
| IMP_CPUECTLR2_EL1      | 3   | 0   | C15 | C1  | 5   | See individual bit resets. | 64-bit | CPU Extended Control Register 2                  |
| IMP_CPUL2DIRTYLNCT_EL1 | 3   | 0   | C15 | C2  | 5   | See individual bit resets. | 64-bit | CPU L2 Dirty Line Count Register                 |
| IMP_CPUPWRCTLR_EL1     | 3   | 0   | C15 | C2  | 7   | See individual bit resets. | 64-bit | CPU Power Control Register                       |
| IMP_ATCR_EL1           | 3   | 0   | C15 | C7  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Translation Control Register (EL1) |
| IMP_CPUACTLR5_EL1      | 3   | 0   | C15 | C8  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 5 (EL1)           |
| IMP_CPUACTLR6_EL1      | 3   | 0   | C15 | C8  | 1   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 6 (EL1)           |
| IMP_CPUACTLR7_EL1      | 3   | 0   | C15 | C8  | 2   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 7 (EL1)           |
| IMP_CPUACTLR10_EL1     | 3   | 0   | C15 | C8  | 3   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 10 (EL1)          |
| IMP_CPUACTLR8_EL1      | 3   | 0   | C15 | C8  | 5   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 8 (EL1)           |
| IMP_CPUACTLR9_EL1      | 3   | 0   | C15 | C8  | 6   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 9 (EL1)           |
| IMP_CPUACTLR11_EL1     | 3   | 0   | C15 | C8  | 7   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register 11 (EL1)          |
| AIDR_EL1               | 3   | 1   | C0  | C0  | 7   | See individual bit resets. | 64-bit | Auxiliary ID Register                            |
| NZCV                   | 3   | 3   | C4  | C2  | 0   | See individual bit resets. | 64-bit | Condition Flags                                  |
| DAIF                   | 3   | 3   | C4  | C2  | 1   | See individual bit resets. | 64-bit | Interrupt Mask Bits                              |

| Name                      | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                           |
|---------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------|
| DIT                       | 3   | 3   | C4  | C2  | 5   | See individual bit resets. | 64-bit | Data Independent Timing                               |
| SSBS                      | 3   | 3   | C4  | C2  | 6   | See individual bit resets. | 64-bit | Speculative Store Bypass Safe                         |
| TCO                       | 3   | 3   | C4  | C2  | 7   | See individual bit resets. | 64-bit | Tag Check Override                                    |
| <a href="#">FPCR</a>      | 3   | 3   | C4  | C4  | 0   | See individual bit resets. | 64-bit | Floating-point Control Register                       |
| FPSR                      | 3   | 3   | C4  | C4  | 1   | See individual bit resets. | 64-bit | Floating-point Status Register                        |
| TPIDR_ELO                 | 3   | 3   | C13 | C0  | 2   | See individual bit resets. | 64-bit | ELO Read/Write Software Thread ID Register            |
| TPIDRRO_ELO               | 3   | 3   | C13 | C0  | 3   | See individual bit resets. | 64-bit | ELO Read-Only Software Thread ID Register             |
| SCXTNUM_ELO               | 3   | 3   | C13 | C0  | 7   | See individual bit resets. | 64-bit | ELO Read/Write Software Context Number                |
| <a href="#">ACTLR_EL2</a> | 3   | 4   | C1  | C0  | 1   | See individual bit resets. | 64-bit | Auxiliary Control Register (EL2)                      |
| <a href="#">HACR_EL2</a>  | 3   | 4   | C1  | C1  | 7   | See individual bit resets. | 64-bit | Hypervisor Auxiliary Control Register                 |
| TTBR0_EL2                 | 3   | 4   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Translation Table Base Register 0 (EL2)               |
| TTBR1_EL2                 | 3   | 4   | C2  | C0  | 1   | See individual bit resets. | 64-bit | Translation Table Base Register 1 (EL2)               |
| TCR_EL2                   | 3   | 4   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Translation Control Register (EL2)                    |
| VTTBR_EL2                 | 3   | 4   | C2  | C1  | 0   | See individual bit resets. | 64-bit | Virtualization Translation Table Base Register        |
| VTCR_EL2                  | 3   | 4   | C2  | C1  | 2   | See individual bit resets. | 64-bit | Virtualization Translation Control Register           |
| VSTTBR_EL2                | 3   | 4   | C2  | C6  | 0   | See individual bit resets. | 64-bit | Virtualization Secure Translation Table Base Register |
| VSTCR_EL2                 | 3   | 4   | C2  | C6  | 2   | See individual bit resets. | 64-bit | Virtualization Secure Translation Control Register    |
| <a href="#">AFSR0_EL2</a> | 3   | 4   | C5  | C1  | 0   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 0 (EL2)               |
| <a href="#">AFSR1_EL2</a> | 3   | 4   | C5  | C1  | 1   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 1 (EL2)               |
| ESR_EL2                   | 3   | 4   | C5  | C2  | 0   | See individual bit resets. | 64-bit | Exception Syndrome Register (EL2)                     |
| TFSR_EL2                  | 3   | 4   | C5  | C6  | 0   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL2)                       |
| FAR_EL2                   | 3   | 4   | C6  | C0  | 0   | See individual bit resets. | 64-bit | Fault Address Register (EL2)                          |
| HPFAR_EL2                 | 3   | 4   | C6  | C0  | 4   | See individual bit resets. | 64-bit | Hypervisor IPA Fault Address Register                 |

| Name           | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                     |
|----------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------------------------|
| MAIR_EL2       | 3   | 4   | C10 | C2  | 0   | See individual bit resets. | 64-bit | Memory Attribute Indirection Register (EL2)                     |
| AMAIR_EL2      | 3   | 4   | C10 | C3  | 0   | See individual bit resets. | 64-bit | Auxiliary Memory Attribute Indirection Register (EL2)           |
| VBAR_EL2       | 3   | 4   | C12 | C0  | 0   | See individual bit resets. | 64-bit | Vector Base Address Register (EL2)                              |
| CONTEXTIDR_EL2 | 3   | 4   | C13 | C0  | 1   | See individual bit resets. | 64-bit | Context ID Register (EL2)                                       |
| TPIDR_EL2      | 3   | 4   | C13 | C0  | 2   | See individual bit resets. | 64-bit | EL2 Software Thread ID Register                                 |
| SCXTNUM_EL2    | 3   | 4   | C13 | C0  | 7   | See individual bit resets. | 64-bit | EL2 Read/Write Software Context Number                          |
| IMP_ATCR_EL2   | 3   | 4   | C15 | C7  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Translation Control Register (EL2)                |
| IMP_AVTCR_EL2  | 3   | 4   | C15 | C7  | 1   | See individual bit resets. | 64-bit | CPU Virtualization Auxiliary Translation Control Register (EL2) |
| ACTLR_EL3      | 3   | 6   | C1  | C0  | 1   | See individual bit resets. | 64-bit | Auxiliary Control Register (EL3)                                |
| SCR_EL3        | 3   | 6   | C1  | C1  | 0   | See individual bit resets. | 64-bit | Secure Configuration Register                                   |
| CPTR_EL3       | 3   | 6   | C1  | C1  | 2   | See individual bit resets. | 64-bit | Architectural Feature Trap Register (EL3)                       |
| MDCR_EL3       | 3   | 6   | C1  | C3  | 1   | See individual bit resets. | 64-bit | Monitor Debug Configuration Register (EL3)                      |
| TTBR0_EL3      | 3   | 6   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Translation Table Base Register 0 (EL3)                         |
| TCR_EL3        | 3   | 6   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Translation Control Register (EL3)                              |
| AFSR0_EL3      | 3   | 6   | C5  | C1  | 0   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 0 (EL3)                         |
| AFSR1_EL3      | 3   | 6   | C5  | C1  | 1   | See individual bit resets. | 64-bit | Auxiliary Fault Status Register 1 (EL3)                         |
| ESR_EL3        | 3   | 6   | C5  | C2  | 0   | See individual bit resets. | 64-bit | Exception Syndrome Register (EL3)                               |
| TFSR_EL3       | 3   | 6   | C5  | C6  | 0   | See individual bit resets. | 64-bit | Tag Fault Status Register (EL3)                                 |
| FAR_EL3        | 3   | 6   | C6  | C0  | 0   | See individual bit resets. | 64-bit | Fault Address Register (EL3)                                    |
| MAIR_EL3       | 3   | 6   | C10 | C2  | 0   | See individual bit resets. | 64-bit | Memory Attribute Indirection Register (EL3)                     |
| AMAIR_EL3      | 3   | 6   | C10 | C3  | 0   | See individual bit resets. | 64-bit | Auxiliary Memory Attribute Indirection Register (EL3)           |
| VBAR_EL3       | 3   | 6   | C12 | C0  | 0   | See individual bit resets. | 64-bit | Vector Base Address Register (EL3)                              |
| RVBAR_EL3      | 3   | 6   | C12 | C0  | 1   | See individual bit resets. | 64-bit | Reset Vector Base Address Register (if EL3 implemented)         |

| Name                    | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                      |
|-------------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------|
| RMR_EL3                 | 3   | 6   | C12 | C0  | 2   | See individual bit resets. | 64-bit | Reset Management Register (EL3)                  |
| TPIDR_EL3               | 3   | 6   | C13 | C0  | 2   | See individual bit resets. | 64-bit | EL3 Software Thread ID Register                  |
| SCXTNUM_EL3             | 3   | 6   | C13 | C0  | 7   | See individual bit resets. | 64-bit | EL3 Read/Write Software Context Number           |
| IMP_CPUL2SDIRTYLNCT_EL3 | 3   | 6   | C15 | C2  | 3   | See individual bit resets. | 64-bit | CPU L2 Secure Dirty Line Count Register          |
| IMP_CPUACTLR_EL3        | 3   | 6   | C15 | C4  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Control Register (EL3)             |
| IMP_ATCR_EL3            | 3   | 6   | C15 | C7  | 0   | See individual bit resets. | 64-bit | CPU Auxiliary Translation Control Register (EL3) |
| IMP_CPUPSELR_EL3        | 3   | 6   | C15 | C8  | 0   | See individual bit resets. | 64-bit | Selected Instruction Private Select Register     |
| IMP_CPUPCR_EL3          | 3   | 6   | C15 | C8  | 1   | See individual bit resets. | 64-bit | Selected Instruction Private Control Register    |
| IMP_CPUPOR_EL3          | 3   | 6   | C15 | C8  | 2   | See individual bit resets. | 64-bit | Selected Instruction Private Opcode Register     |
| IMP_CPUPMR_EL3          | 3   | 6   | C15 | C8  | 3   | See individual bit resets. | 64-bit | Selected Instruction Private Mask Register       |
| IMP_CPUPOR2_EL3         | 3   | 6   | C15 | C8  | 4   | See individual bit resets. | 64-bit | Selected Instruction Private Opcode Register 2   |
| IMP_CPUPMR2_EL3         | 3   | 6   | C15 | C8  | 5   | See individual bit resets. | 64-bit | Selected Instruction Private Mask Register 2     |
| IMP_CPUPFR_EL3          | 3   | 6   | C15 | C8  | 6   | See individual bit resets. | 64-bit | Selected Instruction Private Flag Register       |

## A.2.1 ACTLR\_EL1, Auxiliary Control Register (EL1)

Provides **IMPLEMENTATION DEFINED** configuration and control options for execution at EL1 and EL0.



Note

Arm recommends the contents of this register have no effect on the PE when AArch64-HCR\_EL2.{E2H, TGE} is {1, 1}, and instead the configuration and control fields are provided by the AArch64-ACTLR\_EL2 register. This avoids the need for software to manage the contents of these register when switching between a Guest OS and a Host OS.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-7: AArch64\_actlr\_el1 bit assignments

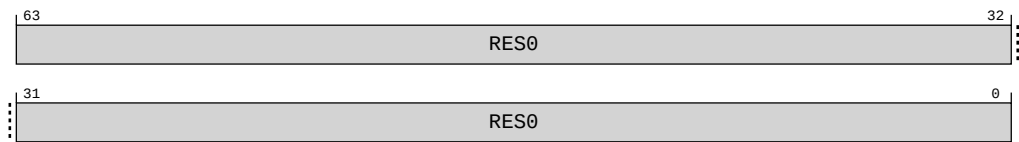


Table A-21: ACTLR\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, ACTLR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0001 | 0b0000 | 0b001 |

MSR ACTLR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0001 | 0b0000 | 0b001 |

Accessibility

MRS <Xt>, ACTLR\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TACR == '1' then
```

```
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ACTLR_EL1;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = ACTLR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = ACTLR_EL1;
```

MSR ACTLR\_EL1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TACR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ACTLR_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    ACTLR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    ACTLR_EL1 = X[t, 64];
```

A.2.2 AFSR0\_EL1, Auxiliary Fault Status Register 0 (EL1)

Provides additional **IMPLEMENTATION DEFINED** fault status information for exceptions taken to EL1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-8: AArch64\_afsr0\_el1 bit assignments

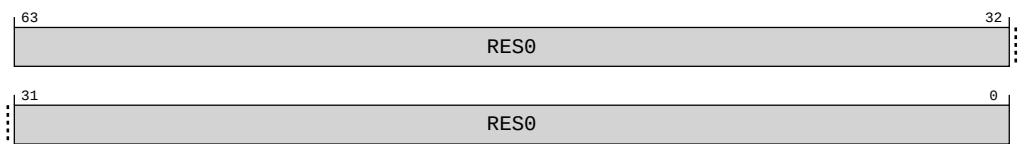


Table A-24: AFSR0\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL3 using the mnemonic AFSR0\_EL1 or AFSR0\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b000 |

MSR AFSR0\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b000 |

MRS <Xt>, AFSR0\_EL12

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b101 | 0b0101 | 0b0001 | 0b000 |

MSR AFSR0\_EL12, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b101 | 0b0101 | 0b0001 | 0b000 |

Accessibility

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL3 using the mnemonic AFSR0\_EL1 or AFSR0\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSR0\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
```



```

elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.AFSRO_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AFSRO_EL1;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AFSRO_EL2;
    else
        X[t, 64] = AFSRO_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AFSRO_EL1;

```

## MSR AFSRO\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.AFSRO_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AFSRO_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AFSRO_EL2 = X[t, 64];
    else
        AFSRO_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AFSRO_EL1 = X[t, 64];

```

## MRS &lt;Xt&gt;, AFSRO\_EL12

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AFSRO_EL1;
    else
        UNDEFINED;
elseif PSTATE.EL == EL3 then
    if EL2Enabled() && HCR_EL2.E2H == '1' then
        X[t, 64] = AFSRO_EL1;
    else
        UNDEFINED;

```

## MSR AFSRO\_EL12, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AFSRO_EL1 = X[t, 64];
    else
        UNDEFINED;
elseif PSTATE.EL == EL3 then

```

```
if EL2Enabled() && HCR_EL2.E2H == '1' then
    AFSR0_EL1 = X[t, 64];
else
    UNDEFINED;
```

A.2.3 AFSR1\_EL1, Auxiliary Fault Status Register 1 (EL1)

Provides additional **IMPLEMENTATION DEFINED** fault status information for exceptions taken to EL1.

Configurations

This register is available in all configurations.

Attributes

Width

64

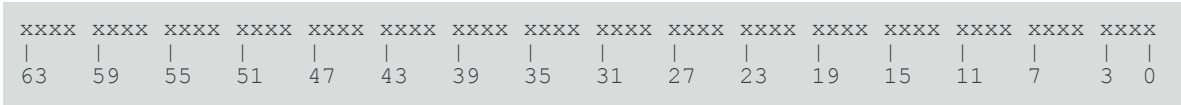
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-9: AArch64\_afsr1\_el1 bit assignments

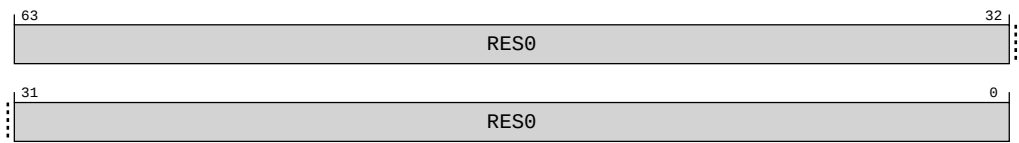


Table A-29: AFSR1\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

## Access

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL3 using the mnemonic AFSR1\_EL1 or AFSR1\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSR1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b001 |

MSR AFSR1\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b001 |

MRS <Xt>, AFSR1\_EL12

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b101 | 0b0101 | 0b0001 | 0b001 |

MSR AFSR1\_EL12, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b101 | 0b0101 | 0b0001 | 0b001 |

## Accessibility

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL3 using the mnemonic AFSR1\_EL1 or AFSR1\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSR1\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.AFSR1_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AFSR1_EL1;
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' then
            X[t, 64] = AFSR1_EL2;
        else
            X[t, 64] = AFSR1_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = AFSR1_EL1;

```

MSR AFSR1\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;

```

```

elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.AFSR1_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AFSR1_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AFSR1_EL2 = X[t, 64];
    else
        AFSR1_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AFSR1_EL1 = X[t, 64];

```

MRS <Xt>, AFSR1\_EL12

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AFSR1_EL1;
    else
        UNDEFINED;
elseif PSTATE.EL == EL3 then
    if EL2Enabled() && HCR_EL2.E2H == '1' then
        X[t, 64] = AFSR1_EL1;
    else
        UNDEFINED;

```

MSR AFSR1\_EL12, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AFSR1_EL1 = X[t, 64];
    else
        UNDEFINED;
elseif PSTATE.EL == EL3 then
    if EL2Enabled() && HCR_EL2.E2H == '1' then
        AFSR1_EL1 = X[t, 64];
    else
        UNDEFINED;

```

## A.2.4 AMAIR\_EL1, Auxiliary Memory Attribute Indirection Register (EL1)

Provides **IMPLEMENTATION DEFINED** memory attributes for the memory regions specified by AArch64-MAIR\_EL1.

### Configurations

This register is available in all configurations.

Attributes

Width

64

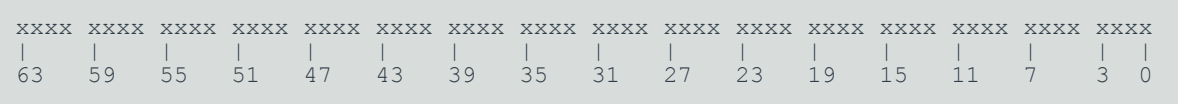
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-10: AArch64\_amair\_el1 bit assignments

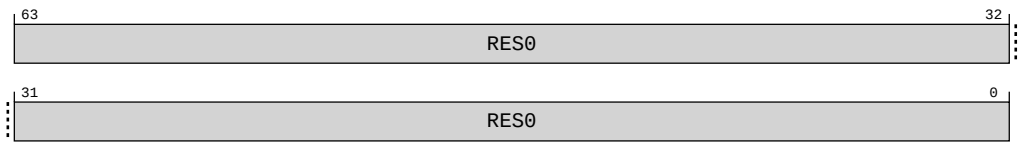


Table A-34: AMAIR\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL3 using the mnemonic AMAIR\_EL1 or AMAIR\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AMAIR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1010 | 0b0011 | 0b000 |

MSR AMAIR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1010 | 0b0011 | 0b000 |

MRS <Xt>, AMAIR\_EL12

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b101 | 0b1010 | 0b0011 | 0b000 |

MSR AMAIR\_EL12, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b101 | 0b1010 | 0b0011 | 0b000 |

## Accessibility

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL3 using the mnemonic AMAIR\_EL1 or AMAIR\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AMAIR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.AMAIR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AMAIR_EL1;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AMAIR_EL2;
    else
        X[t, 64] = AMAIR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = AMAIR_EL1;

```

MSR AMAIR\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.AMAIR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AMAIR_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AMAIR_EL2 = X[t, 64];
    else
        AMAIR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
        AMAIR_EL1 = X[t, 64];

```

MRS <Xt>, AMAIR\_EL12

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[ $\bar{t}$ , 64] = AMAIR_EL1;
    else
        UNDEFINED;
elseif PSTATE.EL == EL3 then
    if EL2Enabled() && HCR_EL2.E2H == '1' then
        X[t, 64] = AMAIR_EL1;
    else
        UNDEFINED;
```

MSR AMAIR\_EL12, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AMAIR_EL1 = X[t, 64];
    else
        UNDEFINED;
elseif PSTATE.EL == EL3 then
    if EL2Enabled() && HCR_EL2.E2H == '1' then
        AMAIR_EL1 = X[t, 64];
    else
        UNDEFINED;
```

A.2.5 LORID\_EL1, LORegionID (EL1)

Indicates the number of LORegions and LORegion descriptors supported by the PE.

Configurations

If no LORegion descriptors are implemented, then the registers AArch64-LORC\_EL1, AArch64-LORN\_EL1, AArch64-LOREA\_EL1, and AArch64-LORSA\_EL1 are RES0.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value



6359555147433935312723191511730



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-11: AArch64\_lorid\_el1 bit assignments

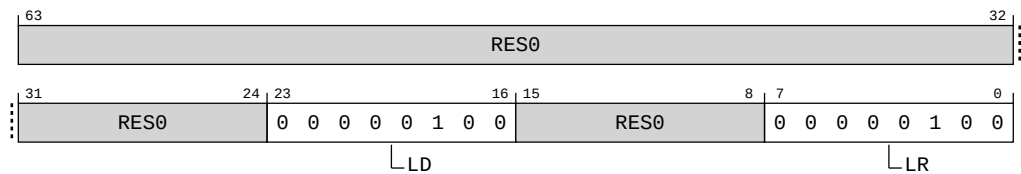


Table A-39: LORID\_EL1 bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                            | Reset |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:24] | RES0 | Reserved                                                                                                                                                                                                                                                                                               | RES0  |
| [23:16] | LD   | Number of LORegion descriptors supported by the PE. This is an 8-bit binary number.<br><br><b>0b000000100</b><br>Four LOR descriptors are supported                                                                                                                                                    | 0x04  |
| [15:8]  | RES0 | Reserved                                                                                                                                                                                                                                                                                               | RES0  |
| [7:0]   | LR   | Number of LORegions supported by the PE. This is an 8-bit binary number.<br><br><b>Note:</b><br>If LORID_EL1 indicates that no LORegions are implemented, then LoadLOAcquire and StoreLORelease will behave as LoadAcquire and StoreRelease.<br><br><b>0b000000100</b><br>Four LORegions are supported | 0x04  |

Access  
MRS <Xt>, LORID\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1010 | 0b0100 | 0b111 |

Accessibility  
MRS <Xt>, LORID\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TLOR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
```



```
elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.LORID_EL1 == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif SCR_EL3.TLOR == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = LORID_EL1;
elseif PSTATE.EL == EL2 then
    if SCR_EL3.TLOR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = LORID_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = LORID_EL1;
```

### A.2.6 IMP\_CPUACTLR\_EL1, CPU Auxiliary Control Register (EL1)

This register contains control bits that affect the CPU behavior

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Generic System Control

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-12: AArch64\_imp\_cpuctlr\_el1 bit assignments



Table A-41: IMP\_CPUACTLR\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_0\_C15\_C1\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b000 |

MSR S3\_0\_C15\_C1\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b000 |

Accessibility

MRS <Xt>, S3\_0\_C15\_C1\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR_EL1;
```

MSR S3\_0\_C15\_C1\_0, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR_EL1 = X[t, 64];
```

```
elseif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    IMP_CPUACTLR_EL1 = X[t, 64];
```

A.2.7 IMP\_CPUACTLR2\_EL1, CPU Auxiliary Control Register 2 (EL1)

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

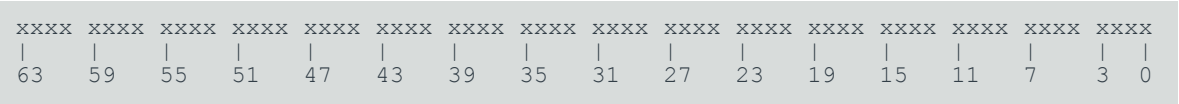
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-13: AArch64\_imp\_cpuactlr2\_el1 bit assignments

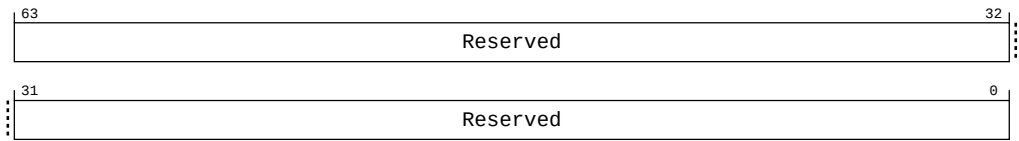


Table A-44: IMP\_CPUACTLR2\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

## Access

MRS <Xt>, S3\_0\_C15\_C1\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b001 |

MSR S3\_0\_C15\_C1\_1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b001 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C1\_1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR2_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR2_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR2_EL1;

```

MSR S3\_0\_C15\_C1\_1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR2_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR2_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUACTLR2_EL1 = X[t, 64];

```

## A.2.8 IMP\_CPUACTLR3\_EL1, CPU Auxiliary Control Register 3 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

Attributes

Width

64

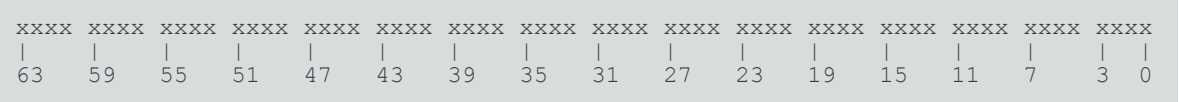
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-14: AArch64\_imp\_cpuctlr3\_el1 bit assignments

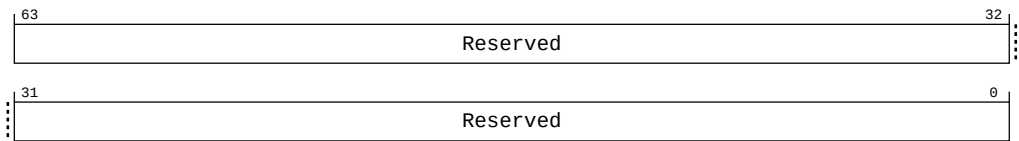


Table A-47: IMP\_CPUACTLR3\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_0\_C15\_C1\_2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b010 |

MSR S3\_0\_C15\_C1\_2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b010 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C1\_2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR3_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR3_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR3_EL1;
```

MSR S3\_0\_C15\_C1\_2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR3_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR3_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUACTLR3_EL1 = X[t, 64];
```

## A.2.9 IMP\_CPUACTLR4\_EL1, CPU Auxiliary Control Register 4 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Generic System Control

#### Access type

See bit descriptions

#### Reset value

```
xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
```



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-15: AArch64\_imp\_cpuctlr4\_el1 bit assignments

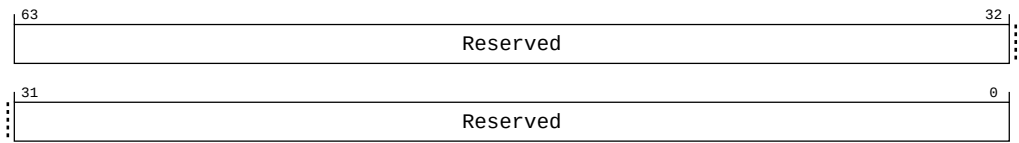


Table A-50: IMP\_CPUACTLR4\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_0\_C15\_C1\_3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b011 |

MSR S3\_0\_C15\_C1\_3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b011 |

Accessibility

MRS <Xt>, S3\_0\_C15\_C1\_3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR4_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR4_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR4_EL1;
```

MSR S3\_0\_C15\_C1\_3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR4_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR4_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUACTLR4_EL1 = X[t, 64];
```

A.2.10 IMP\_CPUECTLR\_EL1, CPU Extended Control Register

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1100 | 0000 | 0000 | 0xxx | 00xx | 0100 | 0000 | 0x11 | 0100 | 0000 | 0101 | 01xx | 0x11 | xx00 | xx00 | 0x00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.



## Bit descriptions

Figure A-16: AArch64\_imp\_cpuctlr\_el1 bit assignments

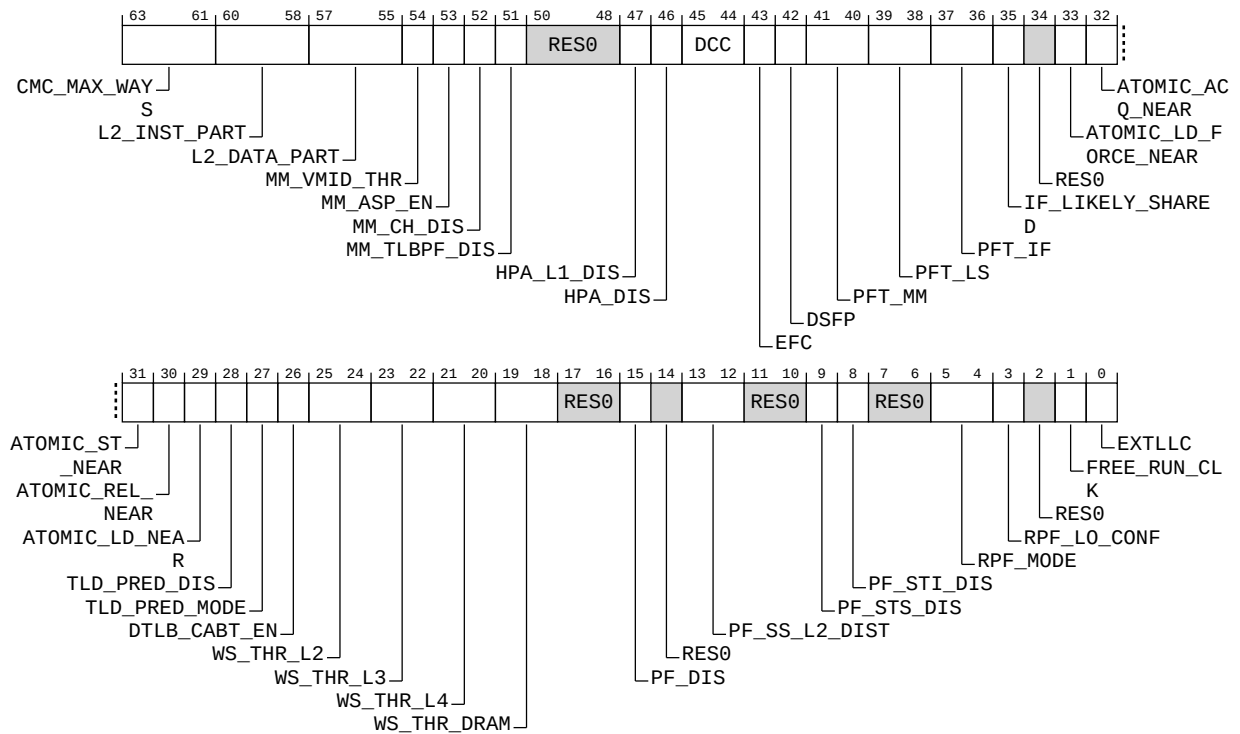


Table A-53: IMP\_CPUCTLRL\_EL1 bit descriptions

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|---------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:61] | CMC_MAX_WAYS | <p>Maximum number of ways of the the L2 used by CMC. The possible values are:</p> <p><b>0b000</b><br/>CMC disabled</p> <p><b>0b001</b><br/>CMC can use 1 way</p> <p><b>0b010</b><br/>CMC can use 2 ways</p> <p><b>0b011</b><br/>CMC can use 3 ways</p> <p><b>0b100</b><br/>CMC can use 4 ways</p> <p><b>0b101</b><br/>CMC can use 5 ways</p> <p><b>0b110</b><br/>CMC can use 6 ways. This is the reset value</p> <p><b>0b111</b><br/>Reserved</p> | 0b110 |

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|---------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [60:58] | L2_INST_PART | <p>Partition the L2 cache for Instruction. The possible values are:</p> <p><b>0b000</b><br/>No ways reserved for instructions. This is the reset value</p> <p><b>0b001</b><br/>Reserve 1 way for instruction. Only instruction fetches can allocate way 7 (2MB) or 11 (3MB)</p> <p><b>0b010</b><br/>Reserve 2 ways for instruction. Only instruction fetches can allocate ways 7:6 (2MB) or 11:10 (3MB)</p> <p><b>0b011</b><br/>Reserve 3 ways for instruction. Only instruction fetches can allocate ways 7:5 (2MB) or 11:9 (3MB)</p> <p><b>0b100</b><br/>Reserve 4 ways for instruction. Only instruction fetches can allocate ways 7:4 (2MB) or 11:8 (3MB)</p> <p><b>0b101</b><br/>Reserve 5 ways for instruction. Only instruction fetches can allocate ways 7:3 (2MB) or 11:7 (3MB)</p> <p><b>0b110</b><br/>Reserve 6 ways for instruction. Only instruction fetches can allocate ways 7:2 (2MB) or 11:6 (3MB)</p> <p><b>0b111</b><br/>Reserved</p> | 0b000 |
| [57:55] | L2_DATA_PART | <p>Reserve L2 capacity for data accesses. The possible values are:</p> <p><b>0b000</b><br/>No ways reserved for data. This is the reset value</p> <p><b>0b001</b><br/>Reserve 1 way for data. Only data accesses can allocate way 0</p> <p><b>0b010</b><br/>Reserve 2 ways for data. Only data accesses can allocate ways 1:0</p> <p><b>0b011</b><br/>Reserve 3 ways for data. Only data accesses can allocate ways 2:0</p> <p><b>0b100</b><br/>Reserve 4 ways for data. Only data accesses can allocate ways 3:0</p> <p><b>0b101</b><br/>Reserve 5 ways for data. Only data accesses can allocate ways 4:0</p> <p><b>0b110</b><br/>Reserve 6 ways for data. Only data accesses can allocate ways 5:0</p> <p><b>0b111</b><br/>Reserved</p>                                                                                                                                                                                                               | 0b000 |

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                    | Reset |
|---------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [54]    | MM_VMID_THR  | <p>VMID filter threshold. The possible values are:</p> <p><b>0b0</b><br/>VMID filter flush after 16 unique VMID allocations to the MMU Translation Cache. This is the default value.</p> <p><b>0b1</b><br/>VMID filter flush after 32 unique VMID allocations to the MMU Translation Cache</p> | 0b0   |
| [53]    | MM_ASP_EN    | <p>Disables allocation of splintered pages in L2 TLB. The possible values are:</p> <p><b>0b0</b><br/>Enables allocation of splintered pages in the L2 TLB. This is the default value.</p> <p><b>0b1</b><br/>Disables allocation of splintered pages in the L2 TLB.</p>                         | 0b0   |
| [52]    | MM_CH_DIS    | <p>Disables use of contiguous hint. The possible values are:</p> <p><b>0b0</b><br/>Enables use of contiguous hint. This is the default value.</p> <p><b>0b1</b><br/>Disables use of contiguous hint.</p>                                                                                       | 0b0   |
| [51]    | MM_TLBPF_DIS | <p>Disables TLB prefetcher. The possible values are:</p> <p><b>0b0</b><br/>Enables TLB prefetcher. This is the default value.</p> <p><b>0b1</b><br/>Disables TLB prefetcher.</p>                                                                                                               | 0b0   |
| [50:48] | RES0         | Reserved                                                                                                                                                                                                                                                                                       | RES0  |
| [47]    | HPA_L1_DIS   | <p>Disables hardware page aggregation in L1 TLBs. The possible values are:</p> <p><b>0b0</b><br/>Enables hardware page aggregation in L1 TLBs. This is the default value.</p> <p><b>0b1</b><br/>Disables hardware page aggregation in L1 TLBs.</p>                                             | 0b0   |
| [46]    | HPA_DIS      | <p>Disable Hardware page aggregation. The possible values are:</p> <p><b>0b0</b><br/>Enables hardware page aggregation. This is the default value.</p> <p><b>0b1</b><br/>Disables hardware page aggregation.</p>                                                                               | 0b0   |
| [45:44] | DCC          | Controls whether evictions of clean cache-lines send data on the CHI interface. Set this based on whether there is a cache on the path to memory. The possible values are:                                                                                                                     | xx    |

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|---------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [43]    | EFC    | <p>Eviction Flush Control (EFC). Controls whether hardware cache flushes and DC CISCW instruction send data when evicting clean and dirty cache line. If it is known that data is likely to be used soon by another core, setting this bit can improve system performance. The possible values are:</p> <p><b>0b0</b></p> <p>Disables cache allocation in downstream caches when hardware cache flushes or DC CISCW instructions evict a clean or dirty cache line. Downstream Snoop Filter Present (DSFP) controls the sending of Evict transactions</p> <p><b>0b1</b></p> <p>Enables cache allocation in downstream caches when hardware cache flushes or DC CISCW instructions evict a clean or dirty cache line. Downstream Cache Control (DCC) controls the sending of data. DSFP controls the sending of Evict transactions.</p> | 0b0   |
| [42]    | DSFP   | <p>Downstream Snoop Filter Present (DSFP). Enables sending Evict transactions on the CHI interface when clean lines are evicted without data. You must enable this if there is at least one snoop filter in the path to memory</p> <p><b>0b0</b></p> <p>Disables sending Evict transactions when clean cachelines are evicted without data</p> <p><b>0b1</b></p> <p>Enables sending of Evict transaction when clean cachelines are evicted without data.</p>                                                                                                                                                                                                                                                                                                                                                                           | 0b1   |
| [41:40] | PFT_MM | <p>DRAM prefetch using PrefetchTgt transactions for tablewalk requests. The possible values are:</p> <p><b>0b00</b></p> <p>Disable prefetchtgt generation for requests from the Memory Management unit (MMU). This is the default value.</p> <p><b>0b01</b></p> <p>Conservatively generate prefetchtgt for cacheable requests from the MMU, always generate for Non-cacheable.</p> <p><b>0b10</b></p> <p>Agressively generate prefetchtgt for cacheable requests from the MMU, always generate for Non-cacheable.</p> <p><b>0b11</b></p> <p>Always generate prefetchtgt for cacheable requests from the MMU, always generate for Non-cacheable.</p>                                                                                                                                                                                    | 0b00  |

| Bits    | Name                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|---------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [39:38] | PFT_LS               | <p>DRAM prefetch using PrefetchTgt transactions for load and store requests. The possible values are:</p> <p><b>0b00</b><br/>Disable prefetchtgt generation for requests from the Load-Store unit (LS). This is the default value.</p> <p><b>0b01</b><br/>Conservatively generate prefetchtgt for cacheable requests from the LS, always generate for Non-cacheable.</p> <p><b>0b10</b><br/>Agressively generate prefetchtgt for cacheable requests from the LS, always generate for Non-cacheable.</p> <p><b>0b11</b><br/>Always generate prefetchtgt for cacheable requests from the LS, always generate for Non-cacheable.</p>           | 0b00  |
| [37:36] | PFT_IF               | <p>DRAM prefetch using PrefetchTgt transactions for instruction fetch requests. The possible values are:</p> <p><b>0b00</b><br/>Disable prefetchtgt generation for requests from the Instruction Fetch unit (IF). This is the default value.</p> <p><b>0b01</b><br/>Conservatively generate prefetchtgt for cacheable requests from the IF, always generate for Non-cacheable.</p> <p><b>0b10</b><br/>Agressively generate prefetchtgt for cacheable requests from the IF, always generate for Non-cacheable.</p> <p><b>0b11</b><br/>Always generate prefetchtgt for cacheable requests from the IF, always generate for Non-cacheable.</p> | 0b00  |
| [35]    | IF_LIKELY_SHARED     | <p>Instruction fetch Shared state control. The possible values are:</p> <p><b>0b0</b><br/>Instruction fetch requests do not assert TXREQ LikelyShared. This is the reset value.</p> <p><b>0b1</b><br/>Instruction fetch requests assert TXREQ LikelyShared and request a SharedClean copy of data.</p>                                                                                                                                                                                                                                                                                                                                      | 0b0   |
| [34]    | RES0                 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [33]    | ATOMIC_LD_FORCE_NEAR | <p>A load atomic (including SWP &amp; CAS) instruction to WB memory will be performed near. The possible values are:</p> <p><b>0b0</b><br/>Load-atomic is near if cache line is already Exclusive, otherwise make far atomic request.</p> <p><b>0b1</b><br/>Load-atomic will be performed near by bringing the line into the L1D Cache. This is the default value.</p>                                                                                                                                                                                                                                                                      | 0b1   |

| Bits | Name            | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [32] | ATOMIC_ACQ_NEAR | <p>An atomic instruction to WB memory with acquire semantics that does not hit in the cache in Exclusive state, may make up to one fill request. The possible values are:</p> <p><b>0b0</b><br/>Acquire-atomic is near if cache line is already Exclusive, otherwise make far atomic request.</p> <p><b>0b1</b><br/>Acquire-atomic will make up to 1 fill request to perform near. This is the default value.</p>  | 0b1   |
| [31] | ATOMIC_ST_NEAR  | <p>A store atomic instruction to WB memory that does not hit in the cache in Exclusive state, may make up to one fill request. The possible values are:</p> <p><b>0b0</b><br/>Store-atomic is near if cache line is already Exclusive, otherwise make far atomic request. This is the default value.</p> <p><b>0b1</b><br/>Store-atomic will make up to 1 fill request to perform near.</p>                        | 0b0   |
| [30] | ATOMIC_REL_NEAR | <p>An atomic instruction to WB memory with release semantics that does not hit in the cache in Exclusive state, may make up to one fill request. The possible values are:</p> <p><b>0b0</b><br/>Release-atomic is near if cache line is already Exclusive, otherwise make far atomic request.</p> <p><b>0b1</b><br/>Release-atomic will make up to 1 fill request to perform near. This is the default value.</p>  | 0b1   |
| [29] | ATOMIC_LD_NEAR  | <p>A load atomic (including SWP &amp; CAS) instruction to WB memory that does not hit in the cache in Exclusive state, may make up to one fill request. The possible values are:</p> <p><b>0b0</b><br/>Load-atomic is near if cache line is already Exclusive, otherwise make far atomic request. This is the default value.</p> <p><b>0b1</b><br/>Load-atomic will make up to 1 fill request to perform near.</p> | 0b0   |
| [28] | TLD_PRED_DIS    | <p>Disable Transient Load Prediction. The possible values are:</p> <p><b>0b0</b><br/>Enables transient load prediction. This is the default value.</p> <p><b>0b1</b><br/>Disables transient load prediction.</p>                                                                                                                                                                                                   | 0b0   |
| [27] | TLD_PRED_MODE   | <p>Aggressive Transient Load Prediction. The possible values are:</p> <p><b>0b0</b><br/>Disables aggressive transient load prediction. This is the default value.</p> <p><b>0b1</b><br/>Enables aggressive transient load prediction.</p>                                                                                                                                                                          | 0b0   |
| [26] | DTLB_CABT_EN    | <p>Enables TLB Conflict Data Abort Exception. The possible values are:</p> <p><b>0b0</b><br/>Disables TLB conflict data abort exception. This is the default value.</p> <p><b>0b1</b><br/>Enables TLB conflict data abort exception.</p>                                                                                                                                                                           | 0b0   |

| Bits    | Name        | Description                                                                                                                                                                                                                                                   | Reset |
|---------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [25:24] | WS_THR_L2   | Threshold for direct stream to L2 cache on store. The possible values are:<br><br><b>0b00</b><br>256B - This is the default value<br><br><b>0b01</b><br>4KB<br><br><b>0b10</b><br>8KB<br><br><b>0b11</b><br>Disables direct stream to L2 cache on store.      | 0b00  |
| [23:22] | WS_THR_L3   | Threshold for direct stream to L3 cache on store. The possible values are:<br><br><b>0b00</b><br>128KB<br><br><b>0b01</b><br>256KB - This is the default value<br><br><b>0b10</b><br>512KB<br><br><b>0b11</b><br>Disables direct stream to L3 cache on store. | 0b01  |
| [21:20] | WS_THR_L4   | Threshold for direct stream to L4 cache on store. The possible values are:<br><br><b>0b00</b><br>256KB<br><br><b>0b01</b><br>512KB - This is the default value<br><br><b>0b10</b><br>1MB<br><br><b>0b11</b><br>Disables direct stream to L4 cache on store.   | 0b01  |
| [19:18] | WS_THR_DRAM | Threshold for direct stream to DRAM on store. The possible values are:<br><br><b>0b00</b><br>512KB<br><br><b>0b01</b><br>1MB - This is the default value<br><br><b>0b10</b><br>2MB<br><br><b>0b11</b><br>Disables direct stream to DRAM on store.             | 0b01  |
| [17:16] | RES0        | Reserved                                                                                                                                                                                                                                                      | RES0  |
| [15]    | PF_DIS      | Disables hardware prefetching. The possible values are:<br><br><b>0b0</b><br>Enables hardware prefetching. This is the default value.<br><br><b>0b1</b><br>Disables hardware prefetching.                                                                     | 0b0   |

| Bits    | Name          | Description                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|---------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [14]    | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [13:12] | PF_SS_L2_DIST | Single cache line stride prefetching L2 distance. The possible values are:<br><br><b>0b00</b><br>22 lines ahead<br><br><b>0b01</b><br>40 lines ahead<br><br><b>0b10</b><br>60 lines ahead<br><br><b>0b11</b><br>Dynamic. This is the default value.                                                                                                                         | 0b11  |
| [11:10] | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [9]     | PF_STS_DIS    | Disable store-stride prefetches. The possible values are:<br><br><b>0b0</b><br>Enables store prefetching. This is the default value.<br><br><b>0b1</b><br>Disables store prefetching.                                                                                                                                                                                       | 0b0   |
| [8]     | PF_STI_DIS    | Disable store prefetches at issue (not overridden by ls_hw_pref_disable). The possible values are:<br><br><b>0b0</b><br>Enables store prefetching. This is the default value.<br><br><b>0b1</b><br>Disable store prefetching.                                                                                                                                               | 0b0   |
| [7:6]   | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [5:4]   | RPF_MODE      | Region prefetcher aggressiveness. The possible values are:<br><br><b>0b00</b><br>Dynamic region prefetch aggressiveness. This is the default value.<br><br><b>0b01</b><br>Conservative region prefetching.<br><br><b>0b10</b><br>Very Conservative region prefetching.<br><br><b>0b11</b><br>Most Conservative region prefetching. This will disable the region prefetcher. | 0b00  |
| [3]     | RPF_LO_CONF   | Region Prefetcher single accesses training behavior. The possible values are:<br><br><b>0b0</b><br>Mostly don't train PHT on single access. This is the default value.<br><br><b>0b1</b><br>Always train the PHT on single access. This results in fewer prefetch requests.                                                                                                 | 0b0   |
| [2]     | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | RES0  |



| Bits | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [1]  | FREE_RUN_CLK | Free-running SRAM clock. The possible values are:<br><br><b>0b0</b><br>Normal clock gate behavior. This is the default value.<br><br><b>0b1</b><br>Enable clock gate override on all the single-cycle RAMs                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0b0   |
| [0]  | EXTLLC       | Internal or external Last-level cache (LLC) in the system. The possible values are:<br><br><b>0b0</b><br>Indicates that an internal Last-level cache is present in the system, and that the DataSource field on the master CHI interface indicates when data is returned from the LLC. This is used to control how the LL_CACHE* PMU events count. This is the default value.<br><br><b>0b1</b><br>Indicates that an external Last-level cache is present in the system, and that the DataSource field on the master CHI interface indicates when data is returned from the LLC. This is used to control how the LL_CACHE* PMU events count. | 0b0   |

## Access

MRS <Xt>, S3\_0\_C15\_C1\_4

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b100 |

MSR S3\_0\_C15\_C1\_4, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b100 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C1\_4

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUECTLR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUECTLR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUECTLR_EL1;

```

MSR S3\_0\_C15\_C1\_4, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && ACTLR_EL2.ECTLREN == '0' then

```

```
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif ACTLR_EL3.ECTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUECTLR_EL1 = X[t, 64];
    elseif PSTATE_EL == EL2 then
        if ACTLR_EL3.ECTLREN == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            IMP_CPUECTLR_EL1 = X[t, 64];
    elseif PSTATE_EL == EL3 then
        IMP_CPUECTLR_EL1 = X[t, 64];
```

### A.2.11 IMP\_CPUECTLR2\_EL1, CPU Extended Control Register 2

This register contains control bits that affect the CPU behavior

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Generic System Control

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | 0001 | 1000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-17: AArch64\_imp\_cpuctlr2\_el1 bit assignments

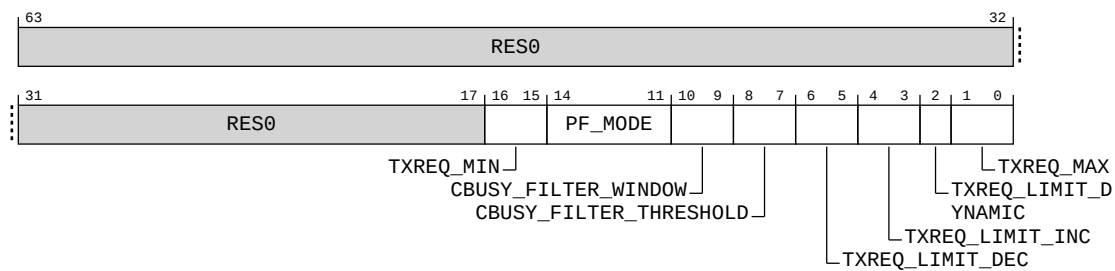


Table A-56: IMP\_CPUECTLR2\_EL1 bit descriptions

| Bits    | Name      | Description                                                                                                                                                                                                                                                                                               | Reset |
|---------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:17] | RES0      | Reserved                                                                                                                                                                                                                                                                                                  | RES0  |
| [16:15] | TXREQ_MIN | Minimum number of TXREQ transactions outstanding from the L2 Transaction Queue.<br>The possible values are:<br><br><b>0b00</b><br>1/4 of L2 TQ size - This is the default value<br><br><b>0b01</b><br>1/8 of L2 TQ size<br><br><b>0b10</b><br>1/16 of L2 TQ size<br><br><b>0b11</b><br>1/32 of L2 TQ size | 0b00  |

| Bits    | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset  |
|---------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [14:11] | PF_MODE | <p>Prefetcher Aggressiveness Modes. With mode 0 representing the most aggressive mode and 3 representing the most conservative mode. The possible values and associated ranges are:</p> <p><b>0b0000</b><br/>Modes [0,0] (statically at the most aggressive mode)</p> <p><b>0b0001</b><br/>Modes [0,1]</p> <p><b>0b0010</b><br/>Modes [0,2]</p> <p><b>0b0011</b><br/>Modes [0,3] - This is the default value.</p> <p><b>0b0100</b><br/>Modes [1,1]</p> <p><b>0b0101</b><br/>Modes [1,2]</p> <p><b>0b0110</b><br/>Modes [1,3]</p> <p><b>0b0111</b><br/>Modes [2,2]</p> <p><b>0b1000</b><br/>Modes [2,3]</p> <p><b>0b1001</b><br/>Modes [3,3] (statically at the most conservative mode)</p> <p><b>0b1010</b><br/>reserved</p> <p><b>0b1011</b><br/>reserved</p> <p><b>0b1100</b><br/>reserved</p> <p><b>0b1101</b><br/>reserved</p> <p><b>0b1110</b><br/>reserved</p> <p><b>0b1111</b><br/>reserved</p> | 0b0011 |

| Bits   | Name                   | Description                                                                                                                                                                                                                                                                             | Reset |
|--------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10:9] | CBUSY_FILTER_WINDOW    | Number of CBusy responses in one sampling window. The possible values are:<br><br><b>0b00</b><br>256 - This is the default value<br><br><b>0b01</b><br>64<br><br><b>0b10</b><br>128<br><br><b>0b11</b><br>512                                                                           | 0b00  |
| [8:7]  | CBUSY_FILTER_THRESHOLD | Fraction of of CBusy responses in the sampling window necessary to be considered a valid sample of that CBusy value. The possible values are:<br><br><b>0b00</b><br>1/16 - This is the default value<br><br><b>0b01</b><br>1/32<br><br><b>0b10</b><br>1/8<br><br><b>0b11</b><br>1/4     | 0b00  |
| [6:5]  | TXREQ_LIMIT_DEC        | Dynamic TXREQ limit decrement. Controls how quickly the dynamic TXREQ limit is decreased when CBusy indicates value of 3. The possible values are:<br><br><b>0b00</b><br>4 - This is the default value<br><br><b>0b01</b><br>8<br><br><b>0b10</b><br>16<br><br><b>0b11</b><br>2         | 0b00  |
| [4:3]  | TXREQ_LIMIT_INC        | Dynamic TXREQ limit increment. Controls how quickly the dynamic TXREQ limit is increased when CBusy indicates values less than 2. The possible values are:<br><br><b>0b00</b><br>4 - This is the default value<br><br><b>0b01</b><br>8<br><br><b>0b10</b><br>16<br><br><b>0b11</b><br>2 | 0b00  |

| Bits  | Name                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [2]   | TXREQ_LIMIT_DYNAMIC | Selects static or dynamic control of TXREQ limit. Dynamic TXREQ limit will adjust based on CBusy responses on RXDAT and RXRSP in the range of the static limit selected by CPUECTLR2_EL1[1:0] and 1/4 of the L2 TQ SIZE. The possible values are:<br><br><b>0b0</b><br>maximum number of TXREQ transactions statically set by CPUECTLR2_EL1[1:0]<br>- This is the default value.<br><br><b>0b1</b><br>maximum number of TXREQ transactions dynamically controlled | 0b0   |
| [1:0] | TXREQ_MAX           | Maximum number of TXREQ transactions outstanding from the L2 Transaction Queue. The possible values are:<br><br><b>0b00</b><br>full L2 TQ size - This is the default value<br><br><b>0b01</b><br>3/4 of L2 TQ size<br><br><b>0b10</b><br>1/2 of L2 TQ size<br><br><b>0b11</b><br>1/4 of L2 TQ size                                                                                                                                                                | 0b00  |

### Access

MRS &lt;Xt&gt;, S3\_0\_C15\_C1\_5

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b101 |

MSR S3\_0\_C15\_C1\_5, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0001 | 0b101 |

### Accessibility

MRS &lt;Xt&gt;, S3\_0\_C15\_C1\_5

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUECTLR2_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUECTLR2_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUECTLR2_EL1;

```

MSR S3\_0\_C15\_C1\_5, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then

```

```

    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && ACTLR_EL2.ECTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif ACTLR_EL3.ECTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUECTLR2_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if ACTLR_EL3.ECTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUECTLR2_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    IMP_CPUECTLR2_EL1 = X[t, 64];

```

## A.2.12 IMP\_CPUL2DIRTYLNCT\_EL1, CPU L2 Dirty Line Count Register

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Generic System Control

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | 0000 | 0000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-18: AArch64\_imp\_cpul2dirtylnct\_el1 bit assignments

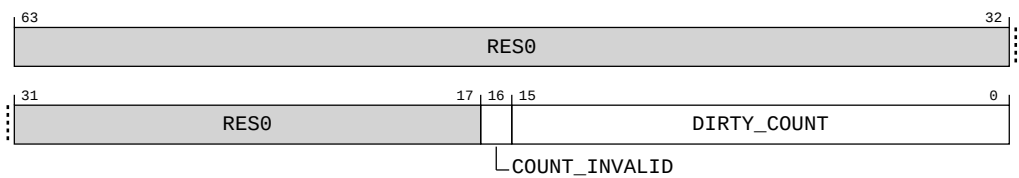


Table A-59: IMP\_CPUL2DIRTYLNCT\_EL1 bit descriptions

| Bits    | Name          | Description                                              | Reset  |
|---------|---------------|----------------------------------------------------------|--------|
| [63:17] | RES0          | Reserved                                                 | RES0   |
| [16]    | COUNT_INVALID | Indicates the dirty count is invalid. Reset value is 'b0 | 0b0    |
| [15:0]  | DIRTY_COUNT   | Number of dirty lines in the L2. Reset value is 'h0000   | 0x0000 |

Access

MRS <Xt>, S3\_0\_C15\_C2\_5

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0010 | 0b101 |

Accessibility

MRS <Xt>, S3\_0\_C15\_C2\_5

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUL2DIRTYLNCT_EL1;
    elseif PSTATE.EL == EL2 then
        X[t, 64] = IMP_CPUL2DIRTYLNCT_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = IMP_CPUL2DIRTYLNCT_EL1;
```

A.2.13 IMP\_CPUPWRCTLR\_EL1, CPU Power Control Register

This register controls various power aspects of the core.

Configurations

This register is available in all configurations.

Attributes

Width

64



Functional group  
Generic System Control

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xx00 | 0000 | xxx0 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-19: AArch64\_imp\_cpupwrctlr\_el1 bit assignments

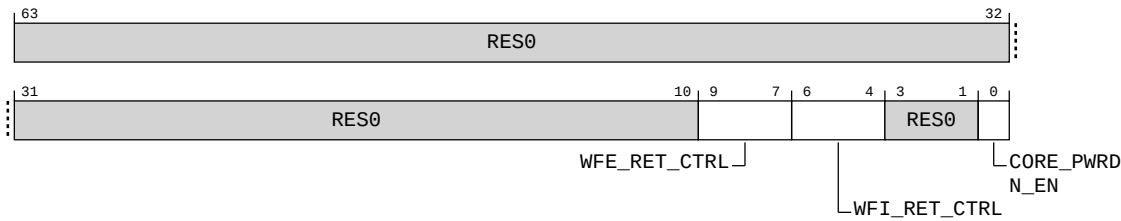


Table A-61: IMP\_CPUPWRCTLR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:10] | RES0 | Reserved    | RES0  |

| Bits  | Name          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [9:7] | WFE_RET_CTRL  | <p>Wait for Event retention control. The possible values are:</p> <p><b>0b000</b><br/>Dynamic retention is disabled.</p> <p><b>0b001</b><br/>2 system counter ticks are required before retention entry.</p> <p><b>0b010</b><br/>8 system counter ticks are required before retention entry.</p> <p><b>0b011</b><br/>32 system counter ticks are required before retention entry.</p> <p><b>0b100</b><br/>64 system counter ticks are required before retention entry.</p> <p><b>0b101</b><br/>128 system counter ticks are required before retention entry.</p> <p><b>0b110</b><br/>256 system counter ticks are required before retention entry.</p> <p><b>0b111</b><br/>512 system counter ticks are required before retention entry.</p>     | 0b000 |
| [6:4] | WFI_RET_CTRL  | <p>Wait for Interrupt retention control. The possible values are:</p> <p><b>0b000</b><br/>Dynamic retention is disabled.</p> <p><b>0b001</b><br/>2 system counter ticks are required before retention entry.</p> <p><b>0b010</b><br/>8 system counter ticks are required before retention entry.</p> <p><b>0b011</b><br/>32 system counter ticks are required before retention entry.</p> <p><b>0b100</b><br/>64 system counter ticks are required before retention entry.</p> <p><b>0b101</b><br/>128 system counter ticks are required before retention entry.</p> <p><b>0b110</b><br/>256 system counter ticks are required before retention entry.</p> <p><b>0b111</b><br/>512 system counter ticks are required before retention entry.</p> | 0b000 |
| [3:1] | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RES0  |
| [0]   | CORE_PWRDN_EN | <p>Indicates to the power controller if the CPU wants to power down when it enters WFE/WFI state. The possible values are:</p> <p><b>0b0</b><br/>CPU does not want to power down when it enters WFE/WFI state.</p> <p><b>0b1</b><br/>CPU wants to power down when it enters WFE/WFI state.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0b0   |

## Access

MRS <Xt>, S3\_0\_C15\_C2\_7

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0010 | 0b111 |

MSR S3\_0\_C15\_C2\_7, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0010 | 0b111 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C2\_7

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUPWRCTLR_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUPWRCTLR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPWRCTLR_EL1;

```

MSR S3\_0\_C15\_C2\_7, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.PWREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.PWREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUPWRCTLR_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.PWREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUPWRCTLR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUPWRCTLR_EL1 = X[t, 64];

```

## A.2.14 IMP\_ATCR\_EL1, CPU Auxiliary Translation Control Register (EL1)

This register contains control bits that affect the CPU behavior.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Functional group

Generic System Control

### Access type

See bit descriptions

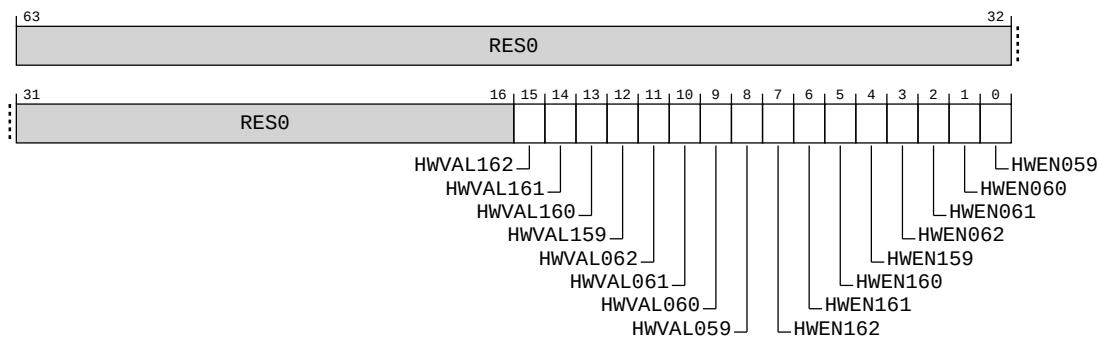
### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-20: AArch64\_imp\_atcr\_el1 bit assignments****Table A-64: IMP\_ATCR\_EL1 bit descriptions**

| Bits    | Name     | Description                                                                                           | Reset |
|---------|----------|-------------------------------------------------------------------------------------------------------|-------|
| [63:16] | RES0     | Reserved                                                                                              | RES0  |
| [15]    | HWVAL162 | Value of PBHA[3] on memory accesses due to translation table walks using TTBR1_EL1 if HWEN162 is set. | 0b0   |
| [14]    | HWVAL161 | Value of PBHA[2] on memory accesses due to translation table walks using TTBR1_EL1 if HWEN161 is set. | 0b0   |
| [13]    | HWVAL160 | Value of PBHA[1] on memory accesses due to translation table walks using TTBR1_EL1 if HWEN160 is set. | 0b0   |
| [12]    | HWVAL159 | Value of PBHA[0] on memory accesses due to translation table walks using TTBR1_EL1 if HWEN159 is set. | 0b0   |
| [11]    | HWVAL062 | Value of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL1 if HWEN062 is set. | 0b0   |
| [10]    | HWVAL061 | Value of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL1 if HWEN061 is set. | 0b0   |
| [9]     | HWVAL060 | Value of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL1 if HWEN060 is set. | 0b0   |
| [8]     | HWVAL059 | Value of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL1 if HWEN059 is set. | 0b0   |

| Bits | Name    | Description                                                                                                                                                  | Reset |
|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7]  | HWEN162 | Enable use of PBHA[3] on memory accesses due to translation table walks using TTBR1_EL1. If this bit is clear, PBHA[3] will be 0 on translation table walks. | 0b0   |
| [6]  | HWEN161 | Enable use of PBHA[2] on memory accesses due to translation table walks using TTBR1_EL1. If this bit is clear, PBHA[2] will be 0 on translation table walks. | 0b0   |
| [5]  | HWEN160 | Enable use of PBHA[1] on memory accesses due to translation table walks using TTBR1_EL1. If this bit is clear, PBHA[1] will be 0 on translation table walks. | 0b0   |
| [4]  | HWEN159 | Enable use of PBHA[0] on memory accesses due to translation table walks using TTBR1_EL1. If this bit is clear, PBHA[0] will be 0 on translation table walks. | 0b0   |
| [3]  | HWEN062 | Enable use of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL1. If this bit is clear, PBHA[3] will be 0 on translation table walks. | 0b0   |
| [2]  | HWEN061 | Enable use of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL1. If this bit is clear, PBHA[2] will be 0 on translation table walks. | 0b0   |
| [1]  | HWEN060 | Enable use of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL1. If this bit is clear, PBHA[1] will be 0 on translation table walks. | 0b0   |
| [0]  | HWEN059 | Enable use of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL1. If this bit is clear, PBHA[0] will be 0 on translation table walks. | 0b0   |

## Access

MRS <Xt>, S3\_0\_C15\_C7\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0111 | 0b000 |

MSR S3\_0\_C15\_C7\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0111 | 0b000 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C7\_0

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_ATCR_EL1;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = IMP_ATCR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = IMP_ATCR_EL1;

```

MSR S3\_0\_C15\_C7\_0, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);

```

```
else
    IMP_ATCR_EL1 = X[t, 64];
elseif PSTATE_EL == EL2 then
    IMP_ATCR_EL1 = X[t, 64];
elseif PSTATE_EL == EL3 then
    IMP_ATCR_EL1 = X[t, 64];
```

A.2.15 IMP\_CPUACTLR5\_EL1, CPU Auxiliary Control Register 5 (EL1)

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

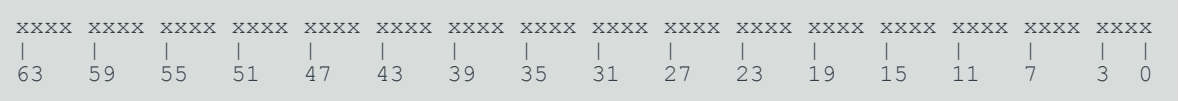
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-21: AArch64\_imp\_cpuctlr5\_el1 bit assignments

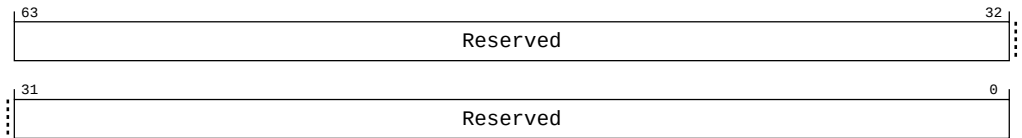


Table A-67: IMP\_CPUACTLR5\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

## Access

MRS <Xt>, S3\_0\_C15\_C8\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b000 |

MSR S3\_0\_C15\_C8\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b000 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_0

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR5_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR5_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR5_EL1;

```

MSR S3\_0\_C15\_C8\_0, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR5_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR5_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUACTLR5_EL1 = X[t, 64];

```

## A.2.16 IMP\_CPUACTLR6\_EL1, CPU Auxiliary Control Register 6 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

Attributes

Width

64

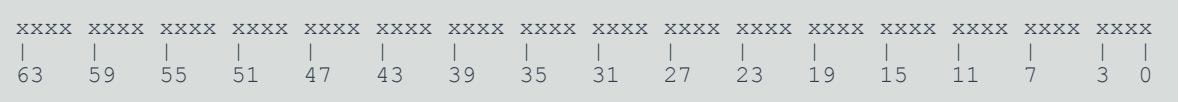
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-22: AArch64\_imp\_cpuctlr6\_el1 bit assignments

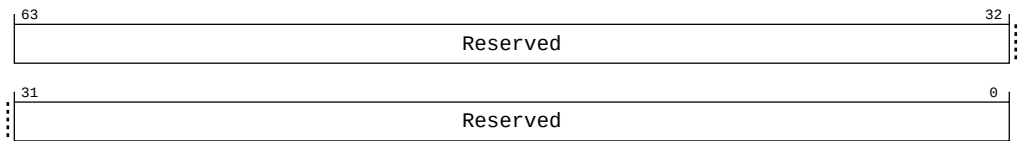


Table A-70: IMP\_CPUACTLR6\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_0\_C15\_C8\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b001 |

MSR S3\_0\_C15\_C8\_1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b001 |



## Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR6_EL1;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = IMP_CPUACTLR6_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = IMP_CPUACTLR6_EL1;
```

MSR S3\_0\_C15\_C8\_1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR6_EL1 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if ACTLR_EL3.ACTLREN == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            IMP_CPUACTLR6_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        IMP_CPUACTLR6_EL1 = X[t, 64];
```

## A.2.17 IMP\_CPUACTLR7\_EL1, CPU Auxiliary Control Register 7 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Generic System Control

#### Access type

See bit descriptions

#### Reset value

```
xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
```



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-23: AArch64\_imp\_cpuctlr7\_el1 bit assignments

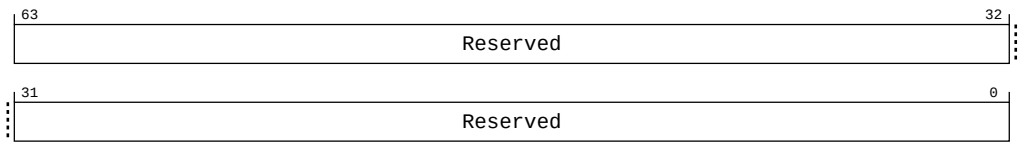


Table A-73: IMP\_CPUACTLR7\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_0\_C15\_C8\_2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b010 |

MSR S3\_0\_C15\_C8\_2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b010 |

Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR7_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR7_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR7_EL1;
```

MSR S3\_0\_C15\_C8\_2, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR7_EL1 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if ACTLR_EL3.ACTLREN == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            IMP_CPUACTLR7_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        IMP_CPUACTLR7_EL1 = X[t, 64];

```

## A.2.18 IMP\_CPUACTLR10\_EL1, CPU Auxiliary Control Register 10 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Generic System Control

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



**Note**

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-24: AArch64\_imp\_cpuctlr10\_el1 bit assignments

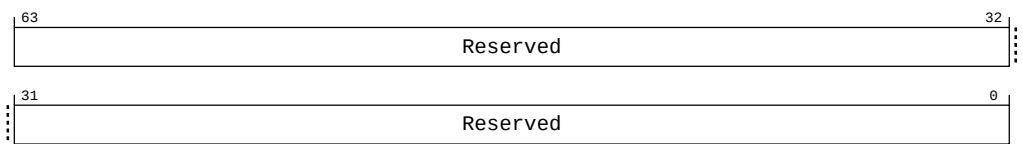


Table A-76: IMP\_CPUACTLR10\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_0\_C15\_C8\_3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b011 |

MSR S3\_0\_C15\_C8\_3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b011 |

Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR10_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR10_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR10_EL1;
```

MSR S3\_0\_C15\_C8\_3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && ACTLR_EL2.ACTLRLEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif ACTLR_EL3.ACTLRLEN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR10_EL1 = X[t, 64];
```

```
elseif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR10_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    IMP_CPUACTLR10_EL1 = X[t, 64];
```

A.2.19 IMP\_CPUACTLR8\_EL1, CPU Auxiliary Control Register 8 (EL1)

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

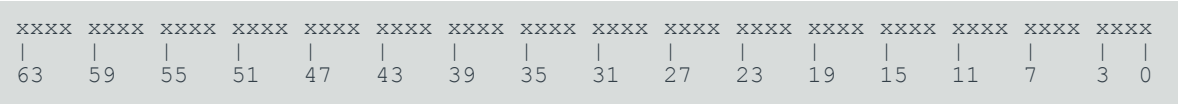
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-25: AArch64\_imp\_cpuactlr8\_el1 bit assignments

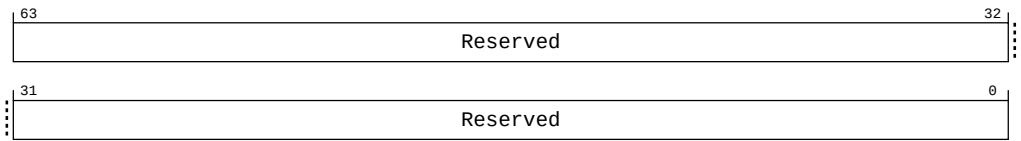


Table A-79: IMP\_CPUACTLR8\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

## Access

MRS <Xt>, S3\_0\_C15\_C8\_5

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b101 |

MSR S3\_0\_C15\_C8\_5, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b101 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_5

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR8_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR8_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR8_EL1;

```

MSR S3\_0\_C15\_C8\_5, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR8_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR8_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUACTLR8_EL1 = X[t, 64];

```

## A.2.20 IMP\_CPUACTLR9\_EL1, CPU Auxiliary Control Register 9 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

Attributes

Width

64

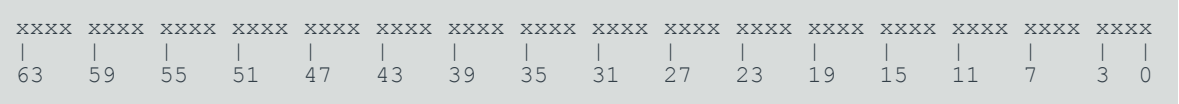
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-26: AArch64\_imp\_cpuctlr9\_el1 bit assignments

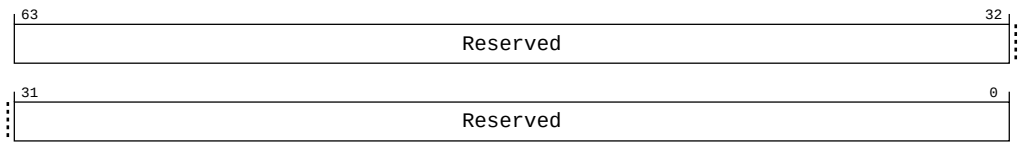


Table A-82: IMP\_CPUACTLR9\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_0\_C15\_C8\_6

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b110 |

MSR S3\_0\_C15\_C8\_6, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b110 |

## Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_6

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR9_EL1;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = IMP_CPUACTLR9_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = IMP_CPUACTLR9_EL1;
```

MSR S3\_0\_C15\_C8\_6, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR9_EL1 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if ACTLR_EL3.ACTLREN == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            IMP_CPUACTLR9_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        IMP_CPUACTLR9_EL1 = X[t, 64];
```

## A.2.21 IMP\_CPUACTLR11\_EL1, CPU Auxiliary Control Register 11 (EL1)

This register contains control bits that affect the CPU behavior

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Generic System Control

#### Access type

See bit descriptions

#### Reset value

```
xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
```





Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-27: AArch64\_imp\_cpuactlr11\_el1 bit assignments

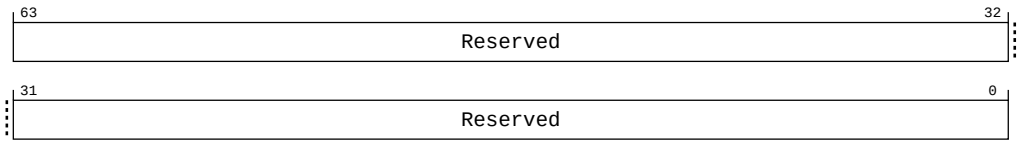


Table A-85: IMP\_CPUACTLR11\_EL1 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_0\_C15\_C8\_7

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b111 |

MSR S3\_0\_C15\_C8\_7, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b1000 | 0b111 |

Accessibility

MRS <Xt>, S3\_0\_C15\_C8\_7

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUACTLR11_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUACTLR11_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR11_EL1;
```

MSR S3\_0\_C15\_C8\_7, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && ACTLR_EL2.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR11_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if ACTLR_EL3.ACTLREN == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        IMP_CPUACTLR11_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_CPUACTLR11_EL1 = X[t, 64];
```

A.2.22 AIDR\_EL1, Auxiliary ID Register

Provides **IMPLEMENTATION DEFINED** identification information.

The value of this register must be interpreted in conjunction with the value of AArch64-MIDR\_EL1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-28: AArch64\_aidr\_el1 bit assignments

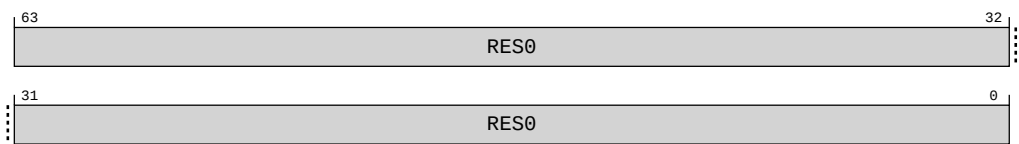


Table A-88: AIDR\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, AIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b001 | 0b0000 | 0b0000 | 0b111 |

Accessibility

MRS <Xt>, AIDR\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.AIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AIDR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = AIDR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AIDR_EL1;
```

A.2.23 FPCR, Floating-point Control Register

Controls floating-point behavior.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group  
Generic System Control

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|---|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |   |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0 |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-29: AArch64\_fpcr bit assignments

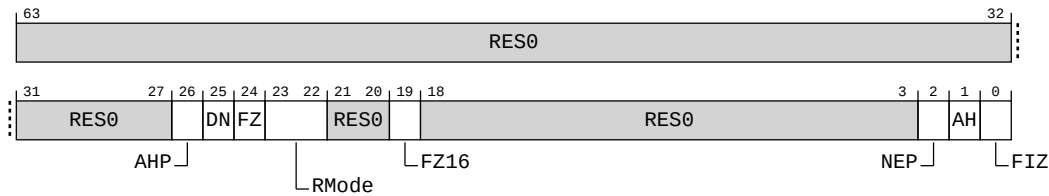


Table A-90: FPCR bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|---------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:27] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                      | RES0  |
| [26]    | AHP  | Alternative half-precision control bit.<br><br><b>0b0</b><br>IEEE half-precision format selected.<br><br><b>0b1</b><br>Alternative half-precision format selected.                                                                                                                                                                                                                            | x     |
| [25]    | DN   | Default NaN use for NaN propagation.<br><br><b>0b0</b><br>NaN operands propagate through to the output of a floating-point operation.<br><br><b>0b1</b><br>Any operation involving one or more NaNs returns the Default NaN.<br><br>This bit has no effect on the output of FABS, FMAX*, FMIN*, and FNEG instructions, and a default NaN is never returned as a result of these instructions. | x     |

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|---------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [24]    | FZ    | <p>Flushing denormalized numbers to zero control bit.</p> <p><b>0b0</b></p> <p>If FPCR.AH is 0, the flushing to zero of single-precision and double-precision denormalized inputs to, and outputs of, floating-point instructions not enabled by this control, but other factors might cause the input denormalized numbers to be flushed to zero.</p> <p>If FPCR.AH is 1, the flushing to zero of single-precision and double-precision denormalized outputs of floating-point instructions not enabled by this control, but other factors might cause the input denormalized numbers to be flushed to zero.</p> <p><b>0b1</b></p> <p>If FPCR.AH is 0, denormalized single-precision and double-precision inputs to, and outputs from, floating-point instructions are flushed to zero.</p> <p>If FPCR.AH is 1, denormalized single-precision and double-precision outputs from floating-point instructions are flushed to zero.</p> | x     |
| [23:22] | RMode | <p>Rounding Mode control field.</p> <p><b>0b00</b></p> <p>Round to Nearest (RN) mode.</p> <p><b>0b01</b></p> <p>Round towards Plus Infinity (RP) mode.</p> <p><b>0b10</b></p> <p>Round towards Minus Infinity (RM) mode.</p> <p><b>0b11</b></p> <p>Round towards Zero (RZ) mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | xx    |
| [21:20] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RES0  |
| [19]    | FZ16  | <p>Flushing denormalized numbers to zero control bit on half-precision data-processing instructions.</p> <p><b>0b0</b></p> <p>For some instructions, this bit disables flushing to zero of inputs and outputs that are half-precision denormalized numbers.</p> <p><b>0b1</b></p> <p>Flushing denormalized numbers to zero enabled.</p> <p>For some instructions that do not convert a half-precision input to a higher precision output, this bit enables flushing to zero of inputs and outputs that are half-precision denormalized numbers.</p>                                                                                                                                                                                                                                                                                                                                                                                   | x     |
| [18:3]  | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RES0  |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [2]  | NEP  | <p>Controls how the output elements other than the lowest element of the vector are determined for Advanced SIMD scalar instructions.</p> <p><b>0b0</b></p> <p>Does not affect how the output elements other than the lowest are determined for Advanced SIMD scalar instructions.</p> <p><b>0b1</b></p> <p>The output elements other than the lowest are taken from the following registers:</p> <ul style="list-style-type: none"> <li>For 3-input scalar versions of the FMLA (by element) and FMLS (by element) instructions, the &lt;Hd&gt;, &lt;Sd&gt;, or &lt;Dd&gt; register.</li> <li>For 3-input versions of the FMADD, FMSUB, FNMADD, and FNMSUB instructions, the &lt;Ha&gt;, &lt;Sa&gt;, or &lt;Da&gt; register.</li> <li>For 2-input scalar versions of the FACGE, FACGT, FCMEQ (register), FCMGE (register), and FCMGT (register) instructions, the &lt;Hm&gt;, &lt;Sm&gt;, or &lt;Dm&gt; register.</li> <li>For 2-input scalar versions of the FABD, FADD (scalar), FDIV (scalar), FMAX (scalar), FMAXNM (scalar), FMIN (scalar), FMINNM (scalar), FMUL (by element), FMUL (scalar), FMULX (by element), FMULX (scalar), FNMUL (scalar), FRECPs, FRSQRTS, and FSUB (scalar) instructions, the &lt;Hn&gt;, &lt;Sn&gt;, or &lt;Dn&gt; register.</li> <li>For 1-input scalar versions of the following instructions, the &lt;Hd&gt;, &lt;Sd&gt;, or &lt;Dd&gt; register: <ul style="list-style-type: none"> <li>The (vector) versions of the FCVTAS, FCVTAU, FCVTMS, FCVTMU, FCVTNS, FCVTNU, FCVTPS, and FCVTPU instructions.</li> <li>The (vector, fixed-point) and (vector, integer) versions of the FCVTZS, FCVTZU, SCVTF, and UCVTF instructions.</li> <li>The (scalar) versions of the FABS, FNEG, FRINT32X, FRINT32Z, FRINT64X, FRINT64Z, FRINTA, FRINTI, FRINTM, FRINTN, FRINTP, FRINTX, FRINTZ, and FSQRT instructions.</li> <li>The (scalar, fixed-point) and (scalar, integer) versions of the SCVTF and UCVTF instructions.</li> <li>The BFCVT, FCVT, FCVTXN, FRECPe, FRECPX, and FRSQRTE instructions.</li> </ul> </li> </ul> | x     |
| [1]  | AH   | <p>Alternate Handling. Controls alternate handling of floating-point numbers.</p> <p><b>0b1</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | x     |
| [0]  | FIZ  | <p>Flush Inputs to Zero. Controls whether single-precision, double-precision and BFloat16 input operands that are denormalized numbers are flushed to zero.</p> <p><b>0b0</b></p> <p>The flushing to zero of single-precision and double-precision denormalized inputs to floating-point instructions not enabled by this control, but other factors might cause the input denormalized numbers to be flushed to zero.</p> <p><b>0b1</b></p> <p>Denormalized single-precision and double-precision inputs to most floating-point instructions flushed to zero.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | x     |

## Access

MRS <Xt>, FPCR

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b0100 | 0b0100 | 0b000 |

MSR FPCR, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b0100 | 0b0100 | 0b000 |

## Accessibility

MRS <Xt>, FPCR

```

if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11') && CPACR_EL1.FPEN != '11' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x00);
        else
            AArch64.SystemAccessTrap(EL1, 0x07);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11' && CPTR_EL2.FPEN != '11' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elsif EL2Enabled() && HCR_EL2.E2H == '1' && CPTR_EL2.FPEN == 'x0' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elsif EL2Enabled() && HCR_EL2.E2H != '1' && CPTR_EL2.TFP == '1' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elsif CPTR_EL3.TFP == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x07);
            else
                X[t, 64] = FPCR;
        elsif PSTATE.EL == EL1 then
            if CPACR_EL1.FPEN == 'x0' then
                AArch64.SystemAccessTrap(EL1, 0x07);
            elsif EL2Enabled() && HCR_EL2.E2H != '1' && CPTR_EL2.TFP == '1' then
                AArch64.SystemAccessTrap(EL2, 0x07);
            elsif EL2Enabled() && HCR_EL2.E2H == '1' && CPTR_EL2.FPEN == 'x0' then
                AArch64.SystemAccessTrap(EL2, 0x07);
            elsif CPTR_EL3.TFP == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x07);
                else
                    X[t, 64] = FPCR;
            elsif PSTATE.EL == EL2 then
                if HCR_EL2.E2H == '0' && CPTR_EL2.TFP == '1' then
                    AArch64.SystemAccessTrap(EL2, 0x07);
                elsif HCR_EL2.E2H == '1' && CPTR_EL2.FPEN == 'x0' then
                    AArch64.SystemAccessTrap(EL2, 0x07);
                elsif CPTR_EL3.TFP == '1' then
                    if Halted() && EDSCR.SDD == '1' then
                        UNDEFINED;
                    else
                        AArch64.SystemAccessTrap(EL3, 0x07);
                    else
                        X[t, 64] = FPCR;
            elsif PSTATE.EL == EL3 then
                if CPTR_EL3.TFP == '1' then
                    AArch64.SystemAccessTrap(EL3, 0x07);
                else
                    X[t, 64] = FPCR;

```

MSR FPCR, <Xt>

```

if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11') && CPACR_EL1.FPEN != '11' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x00);
        else

```

```

        AArch64.SystemAccessTrap(EL1, 0x07);
    elseif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11' && CPTR_EL2.FPEN != '11' then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elseif EL2Enabled() && HCR_EL2.E2H == '1' && CPTR_EL2.FPEN == 'x0' then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elseif EL2Enabled() && HCR_EL2.E2H != '1' && CPTR_EL2.TFP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elseif CPTR_EL3.TFP == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x07);
        else
            FPCR = X[t, 64];
    elseif PSTATE.EL == EL1 then
        if CPACR_EL1.FPEN == 'x0' then
            AArch64.SystemAccessTrap(EL1, 0x07);
        elseif EL2Enabled() && HCR_EL2.E2H != '1' && CPTR_EL2.TFP == '1' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elseif EL2Enabled() && HCR_EL2.E2H == '1' && CPTR_EL2.FPEN == 'x0' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elseif CPTR_EL3.TFP == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x07);
            else
                FPCR = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '0' && CPTR_EL2.TFP == '1' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elseif HCR_EL2.E2H == '1' && CPTR_EL2.FPEN == 'x0' then
            AArch64.SystemAccessTrap(EL2, 0x07);
        elseif CPTR_EL3.TFP == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x07);
            else
                FPCR = X[t, 64];
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TFP == '1' then
            AArch64.SystemAccessTrap(EL3, 0x07);
        else
            FPCR = X[t, 64];

```

## A.2.24 ACTLR\_EL2, Auxiliary Control Register (EL2)

Provides **IMPLEMENTATION DEFINED** configuration and control options for EL2.



Note

Arm recommends the contents of this register are updated to apply to EL0 when AArch64-HCR\_EL2.{E2H, TGE} is {1, 1}, gaining configuration and control fields from the AArch64-ACTLR\_EL1. This avoids the need for software to manage the contents of these register when switching between a Guest OS and a Host OS.

### Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.



Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | 000x | 0xxx | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-30: AArch64\_actlr\_el2 bit assignments

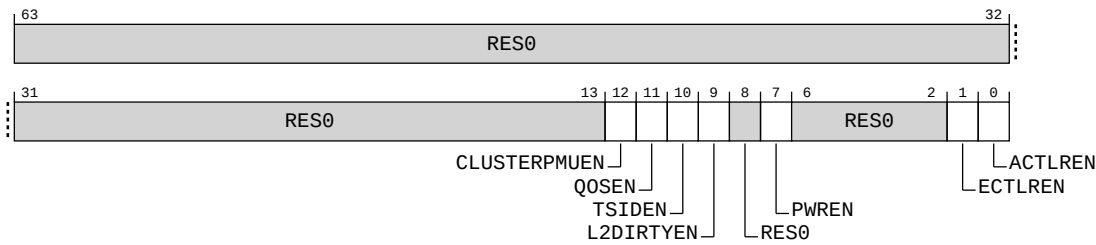


Table A-93: ACTLR\_EL2 bit descriptions

| Bits    | Name         | Description                                                                                                                                                                                                                                                                            | Reset |
|---------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:13] | RES0         | Reserved                                                                                                                                                                                                                                                                               | RES0  |
| [12]    | CLUSTERPMUEN | Performance Management Registers enable. The possible values are:<br><br><b>0b0</b><br>CLUSTERPM* registers are not write-accessible from EL1. This is the reset value.<br><br><b>0b1</b><br>CLUSTERPM* registers are write-accessible from EL1 if they are write-accessible from EL2. | 0b0   |
| [11]    | QOSEN        | CPU Bus QoS Register enable. The possible values are:<br><br><b>0b0</b><br>Register CPUBUSQOS_EL1 is not write-accessible from EL1. This is the reset value.<br><br><b>0b1</b><br>Register CPUBOSQOS_EL1 is write-accessible from EL1 if it is also write-accessible from EL2          | 0b0   |

| Bits  | Name      | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10]  | TSIDEN    | Thread Scheme ID Register enable. The possible values are:<br><br><b>0b0</b><br>Register CLUSTERTHREADSID is not write-accessible from EL1. This is the reset value.<br><br><b>0b1</b><br>Register CLUSTERTHREADSID is write-accessible from EL1 if they are write-accessible from EL2                                                                                                                                                   | 0b0   |
| [9]   | L2DIRTYEN | L2 Dirty Line Count Register enable. The possible values are:<br><br><b>0b0</b><br>Register CPUL2DIRTYLNCT_EL1 is not read-accessible in EL1. This is the reset value.<br><br><b>0b1</b><br>Register CPUL2DIRTYLNCT_EL1 is read-accessible in EL1.                                                                                                                                                                                       | 0b0   |
| [8]   | RES0      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [7]   | PWREN     | Power Control Registers enable. The possible values are:<br><br><b>0b0</b><br>Registers CPUPWRCTLR, CLUSTERPWRCTLR, CLUSTERPWRDN, CLUSTERPWRSTAT, CLUSTERL3HIT and CLUSTERL3MISS are not write accessible from EL1. This is the reset value.<br><br><b>0b1</b><br>Registers CPUPWRCTLR, CLUSTERPWRCTLR, CLUSTERPWRDN, CLUSTERPWRSTAT, CLUSTERL3HIT and CLUSTERL3MISS are write-accessible from EL1 if they are write-accessible from EL2 | 0b0   |
| [6:2] | RES0      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [1]   | ECTLREN   | Extended Control Registers enable. The possible values are:<br><br><b>0b0</b><br>CPUECTLR*_EL1 and CLUSTERECTLR_EL1 are not write-accessible from EL1. This is the reset value.<br><br><b>0b1</b><br>CPUECTLR*_EL1 and CLUSTERECTLR_EL1 are write-accessible from EL1 if they are write-accessible from EL2.                                                                                                                             | 0b0   |
| [0]   | ACTLREN   | Auxiliary Control Registers enable. The possible values are:<br><br><b>0b0</b><br>CPUACTLR*_EL1 and CLUSTERACTLR are not write-accessible from EL1. This is the reset value.<br><br><b>0b1</b><br>CPUACTLR*_EL1 and CLUSTERACTLR are write-accessible from EL1 if they are write-accessible from EL2                                                                                                                                     | 0b0   |

## Access

MRS <Xt>, ACTLR\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0001 | 0b0000 | 0b001 |

MSR ACTLR\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0001 | 0b0000 | 0b001 |

Accessibility

MRS <Xt>, ACTLR\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ACTLR_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ACTLR_EL2;
```

MSR ACTLR\_EL2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    ACTLR_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    ACTLR_EL2 = X[t, 64];
```

A.2.25 HACR\_EL2, Hypervisor Auxiliary Control Register

Controls trapping to EL2 of **IMPLEMENTATION DEFINED** aspects of EL1 or EL0 operation.



Note

Arm recommends that the values in this register do not cause unnecessary traps to EL2 when AArch64-HCR\_EL2.{E2H, TGE} == {1, 1}.

Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

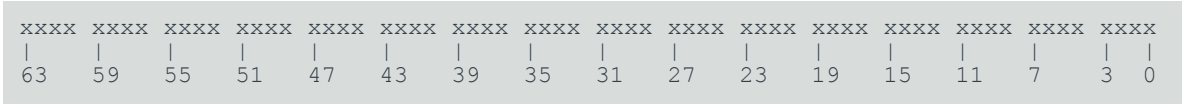
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-31: AArch64\_hacr\_el2 bit assignments

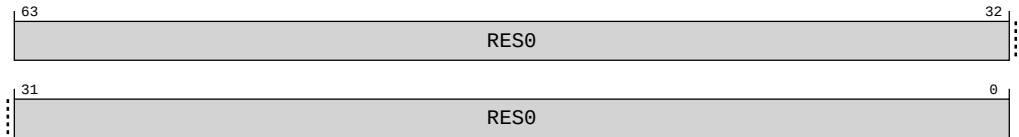


Table A-96: HACR\_EL2 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, HACR\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0001 | 0b0001 | 0b111 |

MSR HACR\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0001 | 0b0001 | 0b111 |

Accessibility

MRS <Xt>, HACR\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    X[t, 64] = HACR_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = HACR_EL2;
```

MSR HACR\_EL2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    HACR_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    HACR_EL2 = X[t, 64];
```

A.2.26 AFSR0\_EL2, Auxiliary Fault Status Register 0 (EL2)

Provides additional **IMPLEMENTATION DEFINED** fault status information for exceptions taken to EL2.

Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-32: AArch64\_afsr0\_el2 bit assignments

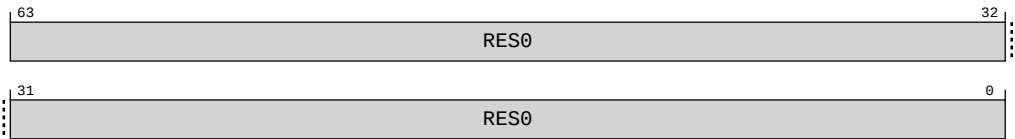


**Table A-99: AFSRO\_EL2 bit descriptions**

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

**Access**

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL2 using the mnemonic AFSRO\_EL2 or AFSRO\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSRO\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0101 | 0b0001 | 0b000 |

MSR AFSRO\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0101 | 0b0001 | 0b000 |

MRS <Xt>, AFSRO\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b000 |

MSR AFSRO\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b000 |

**Accessibility**

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL2 using the mnemonic AFSRO\_EL2 or AFSRO\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSRO\_EL2

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    X[t, 64] = AFSRO_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = AFSRO_EL2;

```

MSR AFSRO\_EL2, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then

```

```

    UNDEFINED;
elseif PSTATE.EL == EL2 then
    AFSR0_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AFSR0_EL2 = X[t, 64];

```

MRS <Xt>, AFSR0\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.AFSR0_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AFSR0_EL1;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AFSR0_EL2;
    else
        X[t, 64] = AFSR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AFSR0_EL1;

```

MSR AFSR0\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.AFSR0_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AFSR0_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AFSR0_EL2 = X[t, 64];
    else
        AFSR0_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AFSR0_EL1 = X[t, 64];

```

## A.2.27 AFSR1\_EL2, Auxiliary Fault Status Register 1 (EL2)

Provides additional **IMPLEMENTATION DEFINED** fault status information for exceptions taken to EL2.

### Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

### Attributes

#### Width

64

Functional group  
Generic System Control

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-33: AArch64\_afsr1\_el2 bit assignments

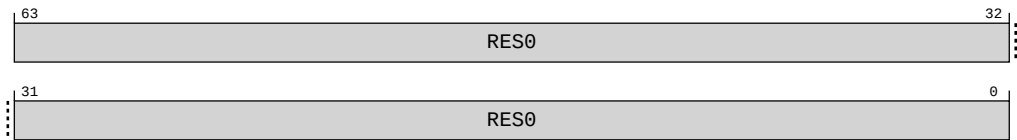


Table A-104: AFSR1\_EL2 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL2 using the mnemonic AFSR1\_EL2 or AFSR1\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSR1\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0101 | 0b0001 | 0b001 |

MSR AFSR1\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b0101 | 0b0001 | 0b001 |

MRS <Xt>, AFSR1\_EL1



| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b001 |

MSR AFSR1\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0001 | 0b001 |

## Accessibility

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL2 using the mnemonic AFSR1\_EL2 or AFSR1\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AFSR1\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    X[t, 64] = AFSR1_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AFSR1_EL2;
```

MSR AFSR1\_EL2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    AFSR1_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AFSR1_EL2 = X[t, 64];
```

MRS <Xt>, AFSR1\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEN == '1' && HFGTR_EL2.AFSR1_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AFSR1_EL1;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AFSR1_EL2;
    else
        X[t, 64] = AFSR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AFSR1_EL1;
```

MSR AFSR1\_EL1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.AFSR1_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AFSR1_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AFSR1_EL2 = X[t, 64];
    else
        AFSR1_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AFSR1_EL1 = X[t, 64];
```

A.2.28 AMAIR\_EL2, Auxiliary Memory Attribute Indirection Register (EL2)

Provides **IMPLEMENTATION DEFINED** memory attributes for the memory regions specified by AArch64-MAIR\_EL2.

Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-34: AArch64\_amair\_el2 bit assignments

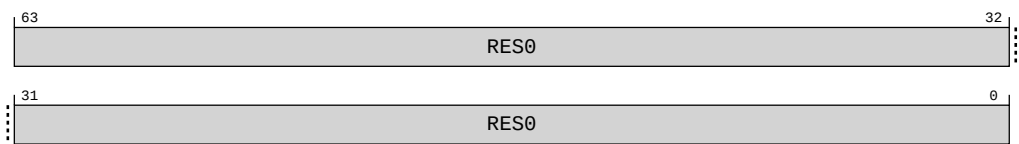


Table A-109: AMAIR\_EL2 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL2 using the mnemonic AMAIR\_EL2 or AMAIR\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AMAIR\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0011 | 0b000 |

MSR AMAIR\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0011 | 0b000 |

MRS <Xt>, AMAIR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1010 | 0b0011 | 0b000 |

MSR AMAIR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1010 | 0b0011 | 0b000 |

Accessibility

When AArch64-HCR\_EL2.E2H is 1, without explicit synchronization, access from EL2 using the mnemonic AMAIR\_EL2 or AMAIR\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

MRS <Xt>, AMAIR\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
```

```

elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    X[t, 64] = AMAIR_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMAIR_EL2;

```

## MSR AMAIR\_EL2, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    AMAIR_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AMAIR_EL2 = X[t, 64];

```

## MRS &lt;Xt&gt;, AMAIR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.AMAIR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = AMAIR_EL1;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AMAIR_EL2;
    else
        X[t, 64] = AMAIR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMAIR_EL1;

```

## MSR AMAIR\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.AMAIR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AMAIR_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AMAIR_EL2 = X[t, 64];
    else
        AMAIR_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    AMAIR_EL1 = X[t, 64];

```

A.2.29 IMP\_ATCR\_EL2, CPU Auxiliary Translation Control Register (EL2)

This register contains control bits that affect the CPU behavior.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-35: AArch64\_imp\_atcr\_el2 bit assignments

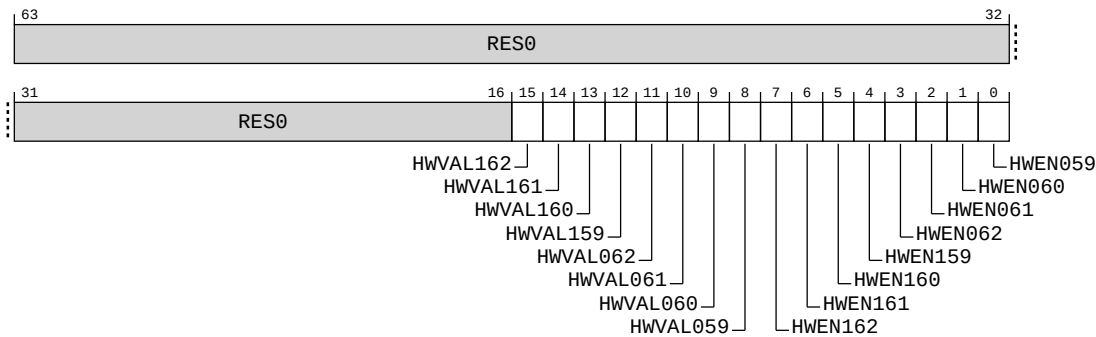


Table A-114: IMP\_ATCR\_EL2 bit descriptions

| Bits    | Name     | Description                                                                                           | Reset |
|---------|----------|-------------------------------------------------------------------------------------------------------|-------|
| [63:16] | RES0     | Reserved                                                                                              | RES0  |
| [15]    | HWVAL162 | Value of PBHA[3] on memory accesses due to translation table walks using TTBR1_EL2 if HWEN162 is set. | 0b0   |
| [14]    | HWVAL161 | Value of PBHA[2] on memory accesses due to translation table walks using TTBR1_EL2 if HWEN161 is set. | 0b0   |

| Bits | Name     | Description                                                                                                                                                  | Reset |
|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [13] | HWVAL160 | Value of PBHA[1] on memory accesses due to translation table walks using TTBR1_EL2 if HWEN160 is set.                                                        | 0b0   |
| [12] | HWVAL159 | Value of PBHA[0] on memory accesses due to translation table walks using TTBR1_EL2 if HWEN159 is set.                                                        | 0b0   |
| [11] | HWVAL062 | Value of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN062 is set.                                                        | 0b0   |
| [10] | HWVAL061 | Value of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN061 is set.                                                        | 0b0   |
| [9]  | HWVAL060 | Value of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN060 is set.                                                        | 0b0   |
| [8]  | HWVAL059 | Value of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN059 is set.                                                        | 0b0   |
| [7]  | HWEN162  | Enable use of PBHA[3] on memory accesses due to translation table walks using TTBR1_EL2. If this bit is clear, PBHA[3] will be 0 on translation table walks. | 0b0   |
| [6]  | HWEN161  | Enable use of PBHA[2] on memory accesses due to translation table walks using TTBR1_EL2. If this bit is clear, PBHA[2] will be 0 on translation table walks. | 0b0   |
| [5]  | HWEN160  | Enable use of PBHA[1] on memory accesses due to translation table walks using TTBR1_EL2. If this bit is clear, PBHA[1] will be 0 on translation table walks. | 0b0   |
| [4]  | HWEN159  | Enable use of PBHA[0] on memory accesses due to translation table walks using TTBR1_EL2. If this bit is clear, PBHA[0] will be 0 on translation table walks. | 0b0   |
| [3]  | HWEN062  | Enable use of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[3] will be 0 on translation table walks. | 0b0   |
| [2]  | HWEN061  | Enable use of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[2] will be 0 on translation table walks. | 0b0   |
| [1]  | HWEN060  | Enable use of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[1] will be 0 on translation table walks. | 0b0   |
| [0]  | HWEN059  | Enable use of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[0] will be 0 on translation table walks. | 0b0   |

## Access

MRS <Xt>, S3\_4\_C15\_C7\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1111 | 0b0111 | 0b000 |

MSR S3\_4\_C15\_C7\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1111 | 0b0111 | 0b000 |

## Accessibility

MRS <Xt>, S3\_4\_C15\_C7\_0

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_ATCR_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_ATCR_EL2;

```

MSR S3\_4\_C15\_C7\_0, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    IMP_ATCR_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    IMP_ATCR_EL2 = X[t, 64];
```

A.2.30 IMP\_AVTCR\_EL2, CPU Virtualization Auxiliary Translation Control Register (EL2)

This register contains control bits that affect the CPU behavior.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-36: AArch64\_imp\_avtcr\_el2 bit assignments**

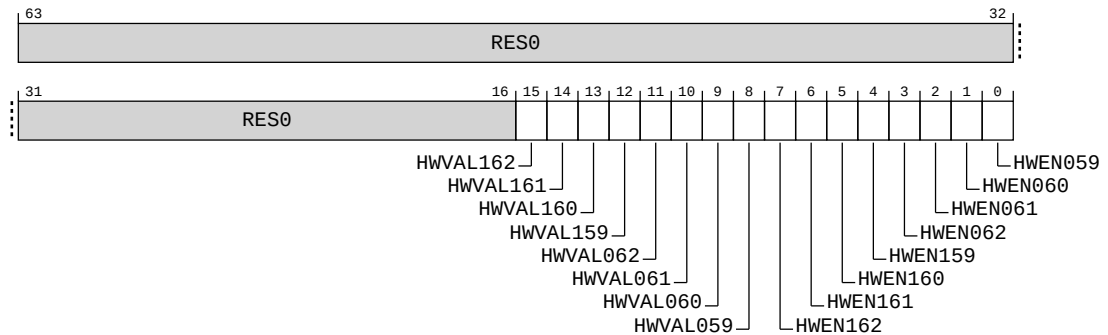


**Table A-117: IMP\_AVTCR\_EL2 bit descriptions**

| Bits    | Name     | Description                                                                                                                                                   | Reset |
|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:16] | RES0     | Reserved                                                                                                                                                      | RES0  |
| [15]    | HWVAL162 | Value of PBHA[3] on memory accesses due to translation table walks using VSTTBR_EL2 if HWEN162 is set.                                                        | 0b0   |
| [14]    | HWVAL161 | Value of PBHA[2] on memory accesses due to translation table walks using VSTTBR_EL2 if HWEN161 is set.                                                        | 0b0   |
| [13]    | HWVAL160 | Value of PBHA[1] on memory accesses due to translation table walks using VSTTBR_EL2 if HWEN160 is set.                                                        | 0b0   |
| [12]    | HWVAL159 | Value of PBHA[0] on memory accesses due to translation table walks using VSTTBR_EL2 if HWEN159 is set.                                                        | 0b0   |
| [11]    | HWVAL062 | Value of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN062 is set.                                                         | 0b0   |
| [10]    | HWVAL061 | Value of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN061 is set.                                                         | 0b0   |
| [9]     | HWVAL060 | Value of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN060 is set.                                                         | 0b0   |
| [8]     | HWVAL059 | Value of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL2 if HWEN059 is set.                                                         | 0b0   |
| [7]     | HWEN162  | Enable use of PBHA[3] on memory accesses due to translation table walks using VSTTBR_EL2. If this bit is clear, PBHA[3] will be 0 on translation table walks. | 0b0   |
| [6]     | HWEN161  | Enable use of PBHA[2] on memory accesses due to translation table walks using VSTTBR_EL2. If this bit is clear, PBHA[2] will be 0 on translation table walks. | 0b0   |
| [5]     | HWEN160  | Enable use of PBHA[1] on memory accesses due to translation table walks using VSTTBR_EL2. If this bit is clear, PBHA[1] will be 0 on translation table walks. | 0b0   |
| [4]     | HWEN159  | Enable use of PBHA[0] on memory accesses due to translation table walks using VSTTBR_EL2. If this bit is clear, PBHA[0] will be 0 on translation table walks. | 0b0   |
| [3]     | HWEN062  | Enable use of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[3] will be 0 on translation table walks.  | 0b0   |
| [2]     | HWEN061  | Enable use of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[2] will be 0 on translation table walks.  | 0b0   |
| [1]     | HWEN060  | Enable use of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[1] will be 0 on translation table walks.  | 0b0   |
| [0]     | HWEN059  | Enable use of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL2. If this bit is clear, PBHA[0] will be 0 on translation table walks.  | 0b0   |



Access

MRS <Xt>, S3\_4\_C15\_C7\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1111 | 0b0111 | 0b001 |

MSR S3\_4\_C15\_C7\_1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1111 | 0b0111 | 0b001 |

Accessibility

MRS <Xt>, S3\_4\_C15\_C7\_1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_AVTCR_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_AVTCR_EL2;
```

MSR S3\_4\_C15\_C7\_1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    IMP_AVTCR_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    IMP_AVTCR_EL2 = X[t, 64];
```

A.2.31 ACTLR\_EL3, Auxiliary Control Register (EL3)

Provides **IMPLEMENTATION DEFINED** configuration and control options for EL3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

## Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | 000x | 0xxx | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure A-37: AArch64\_actlr\_el3 bit assignments

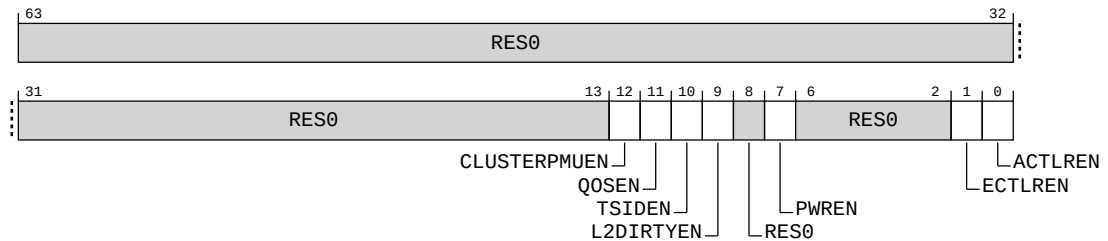


Table A-120: ACTLR\_EL3 bit descriptions

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                            | Reset |
|---------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:13] | RES0         | Reserved                                                                                                                                                                                                                                                                                               | RES0  |
| [12]    | CLUSTERPMUEN | Performance Management Registers enable. The possible values are:<br><br><b>0b0</b><br>CLUSTERPM* registers are not write-accessible from EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>CLUSTERPM* registers are write-accessible from EL2 and EL1 if they are write-accessible from EL2. | 0b0   |
| [11]    | QOSEN        | CPU Bus QoS Register enable. The possible values are:<br><br><b>0b0</b><br>Register CPUBUSQOS_EL1 is not write-accessible EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>Register CPUBOSQOS_EL1 is write-accessible from EL2, and EL1 (if write-accessible from EL2)                       | 0b0   |
| [10]    | TSIDEN       | Thread Scheme ID Register enable. The possible values are:<br><br><b>0b0</b><br>Register CLUSTERTHREADSID is not write-accessible from EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>Register CLUSTERTHREADSID is write-accessible from EL2 and EL1 if they are write-accessible from EL2 | 0b0   |

| Bits  | Name      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [9]   | L2DIRTYEN | L2 Dirty Line Count Register enable. The possible values are:<br><br><b>0b0</b><br>Register CPUL2DIRTYLNCT_EL1 is not read-accessible in EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>Register CPUL2DIRTYLNCT_EL1 is read-accessible in EL2 and EL1.                                                                                                                                                                                       | 0b0   |
| [8]   | RES0      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [7]   | PWREN     | Power Control Registers enable. The possible values are:<br><br><b>0b0</b><br>Registers CPUPWRCTLR, CLUSTERPWRCTLR, CLUSTERPWRDN, CLUSTERPWRSTAT, CLUSTERL3HIT and CLUSTERL3MISS are not write accessible from EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>Registers CPUPWRCTLR, CLUSTERPWRCTLR, CLUSTERPWRDN, CLUSTERPWRSTAT, CLUSTERL3HIT and CLUSTERL3MISS are write-accessible from EL2 and EL1 if they are write-accessible from EL2 | 0b0   |
| [6:2] | RES0      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [1]   | ECTLREN   | Extended Control Registers enable. The possible values are:<br><br><b>0b0</b><br>CPUECTLR*_EL2 and EL1 and CLUSTERECTLR_EL2 and EL1 are not write-accessible from EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>CPUECTLR*_EL2 and EL1 and CLUSTERECTLR_EL2 and EL1 are write-accessible from EL2 and EL1 if they are write-accessible from EL2.                                                                                             | 0b0   |
| [0]   | ACTLREN   | Auxiliary Control Registers enable. The possible values are:<br><br><b>0b0</b><br>CPUACTLR*_EL2 and EL1 and CLUSTERACTLR are not write-accessible from EL2 and EL1. This is the reset value.<br><br><b>0b1</b><br>CPUACTLR*_EL2 and EL1 and CLUSTERACTLR are write-accessible from EL2 and EL1 if they are write-accessible from EL2                                                                                                                     | 0b0   |

## Access

MRS <Xt>, ACTLR\_EL3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b0001 | 0b0000 | 0b001 |

MSR ACTLR\_EL3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b0001 | 0b0000 | 0b001 |

Accessibility

MRS <Xt>, ACTLR\_EL3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ACTLR_EL3;
```

MSR ACTLR\_EL3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    ACTLR_EL3 = X[t, 64];
```

A.2.32 AFSR0\_EL3, Auxiliary Fault Status Register 0 (EL3)

Provides additional **IMPLEMENTATION DEFINED** fault status information for exceptions taken to EL3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-38: AArch64\_afsr0\_el3 bit assignments

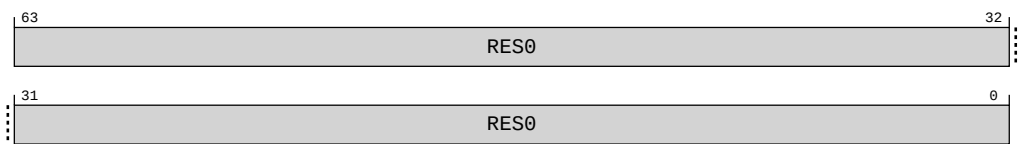


Table A-123: AFSR0\_EL3 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, AFSR0\_EL3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b0101 | 0b0001 | 0b000 |

MSR AFSR0\_EL3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b0101 | 0b0001 | 0b000 |

Accessibility

MRS <Xt>, AFSR0\_EL3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = AFSR0_EL3;
```

MSR AFSR0\_EL3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    AFSR0_EL3 = X[t, 64];
```

A.2.33 AFSR1\_EL3, Auxiliary Fault Status Register 1 (EL3)

Provides additional **IMPLEMENTATION DEFINED** fault status information for exceptions taken to EL3.

Configurations

This register is available in all configurations.

Attributes

Width

64

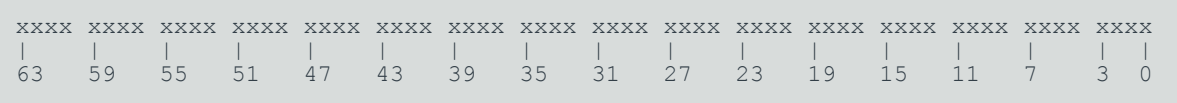
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-39: AArch64\_afsr1\_el3 bit assignments

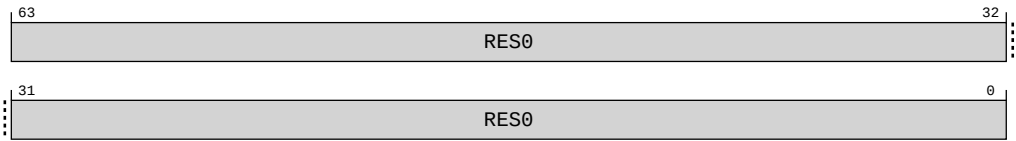


Table A-126: AFSR1\_EL3 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, AFSR1\_EL3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b0101 | 0b0001 | 0b001 |

MSR AFSR1\_EL3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b0101 | 0b0001 | 0b001 |

Accessibility

MRS <Xt>, AFSR1\_EL3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = AFSR1_EL3;
```

MSR AFSR1\_EL3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    AFSR1_EL3 = X[t, 64];
```

A.2.34 AMAIR\_EL3, Auxiliary Memory Attribute Indirection Register (EL3)

Provides **IMPLEMENTATION DEFINED** memory attributes for the memory regions specified by AArch64-MAIR\_EL3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-40: AArch64\_amair\_el3 bit assignments

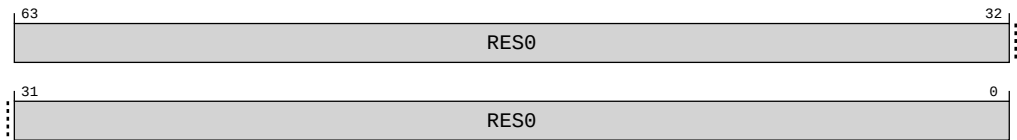


Table A-129: AMAIR\_EL3 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, AMAIR\_EL3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1010 | 0b0011 | 0b000 |

MSR AMAIR\_EL3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1010 | 0b0011 | 0b000 |

Accessibility

MRS <Xt>, AMAIR\_EL3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMAIR_EL3;
```

MSR AMAIR\_EL3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
```



```
elseif PSTATE.EL == EL3 then
    AMAIR_EL3 = X[t, 64];
```

A.2.35 RMR\_EL3, Reset Management Register (EL3)

A write to the register at EL3 can request a Warm reset.

Configurations

When EL3 is implemented:

- If EL3 can use all Execution states then this register must be implemented.
- In a AArch64 only implementation it is IMPLEMENTATION DEFINED whether the register is implemented.

Otherwise, direct accesses to RMR\_EL3 are UNDEFINED.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xx01 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-41: AArch64\_rmr\_el3 bit assignments

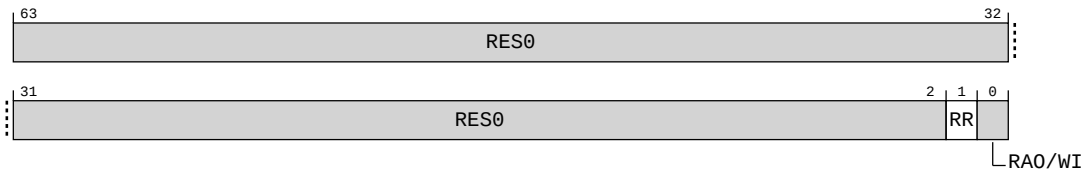


**Table A-132: RMR\_EL3 bit descriptions**

| Bits   | Name          | Description                                                 | Reset         |
|--------|---------------|-------------------------------------------------------------|---------------|
| [63:2] | <b>RES0</b>   | Reserved                                                    | <b>RES0</b>   |
| [1]    | <b>RR</b>     | Reset Request. Setting this bit to 1 requests a Warm reset. | 0b0           |
| [0]    | <b>RAO/WI</b> | Reserved                                                    | <b>RAO/WI</b> |

**Access**

MRS &lt;Xt&gt;, RMR\_EL3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1100 | 0b0000 | 0b010 |

MSR RMR\_EL3, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1100 | 0b0000 | 0b010 |

**Accessibility**

MRS &lt;Xt&gt;, RMR\_EL3

```

if PSTATE.EL == EL3 then
    X[t, 64] = RMR_EL3;
else
    UNDEFINED;

```

MSR RMR\_EL3, &lt;Xt&gt;

```

if PSTATE.EL == EL3 then
    RMR_EL3 = X[t, 64];
else
    UNDEFINED;

```

**A.2.36 IMP\_CPUL2SDIRTYLNCT\_EL3, CPU L2 Secure Dirty Line Count Register**

This register contains control bits that affect the CPU behavior

**Configurations**

This register is available in all configurations.

**Attributes****Width**

64

**Functional group**

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | 0000 | 0000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-42: AArch64\_imp\_cpul2dirtylnct\_el3 bit assignments

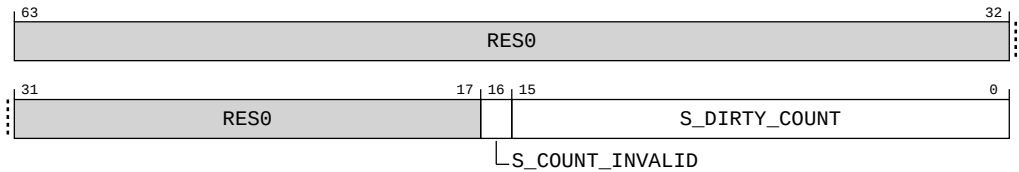


Table A-135: IMP\_CPUL2SDIRTYLNCT\_EL3 bit descriptions

| Bits    | Name            | Description                                                     | Reset  |
|---------|-----------------|-----------------------------------------------------------------|--------|
| [63:17] | RES0            | Reserved                                                        | RES0   |
| [16]    | S_COUNT_INVALID | Indicates the secure dirty count is invalid. Reset value is 'b0 | 0b0    |
| [15:0]  | S_DIRTY_COUNT   | Number of dirty secure lines in the L2. Reset value is 'h0000   | 0x0000 |

Access

MRS <Xt>, S3\_6\_C15\_C2\_3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0010 | 0b011 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C2\_3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUL2SDIRTYLNCT_EL3;
```

A.2.37 IMP\_CPUACTLR\_EL3, CPU Auxiliary Control Register (EL3)

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-43: AArch64\_imp\_cpuctlr\_el3 bit assignments

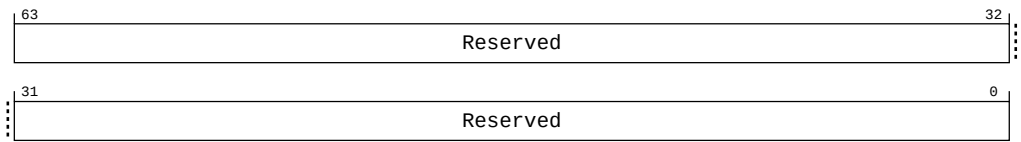


Table A-137: IMP\_CPUACTLR\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_6\_C15\_C4\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0100 | 0b000 |

MSR S3\_6\_C15\_C4\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0100 | 0b000 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C4\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUACTLR_EL3;
```

MSR S3\_6\_C15\_C4\_0, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    IMP_CPUACTLR_EL3 = X[t, 64];
```

A.2.38 IMP\_ATCR\_EL3, CPU Auxiliary Translation Control Register (EL3)

This register contains control bits that affect the CPU behavior.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | xxxx | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-44: AArch64\_imp\_atcr\_el3 bit assignments**

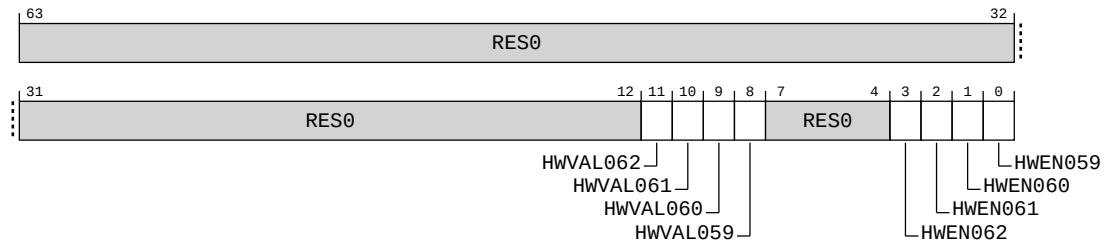


**Table A-140: IMP\_ATCR\_EL3 bit descriptions**

| Bits    | Name        | Description                                                                                                                                                  | Reset       |
|---------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [63:12] | <b>RES0</b> | Reserved                                                                                                                                                     | <b>RES0</b> |
| [11]    | HWVAL062    | Value of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL3 if HWEN062 is set.                                                        | 0b0         |
| [10]    | HWVAL061    | Value of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL3 if HWEN061 is set.                                                        | 0b0         |
| [9]     | HWVAL060    | Value of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL3 if HWEN060 is set.                                                        | 0b0         |
| [8]     | HWVAL059    | Value of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL3 if HWEN059 is set.                                                        | 0b0         |
| [7:4]   | <b>RES0</b> | Reserved                                                                                                                                                     | <b>RES0</b> |
| [3]     | HWEN062     | Enable use of PBHA[3] on memory accesses due to translation table walks using TTBR0_EL3. If this bit is clear, PBHA[3] will be 0 on translation table walks. | 0b0         |
| [2]     | HWEN061     | Enable use of PBHA[2] on memory accesses due to translation table walks using TTBR0_EL3. If this bit is clear, PBHA[2] will be 0 on translation table walks. | 0b0         |
| [1]     | HWEN060     | Enable use of PBHA[1] on memory accesses due to translation table walks using TTBR0_EL3. If this bit is clear, PBHA[1] will be 0 on translation table walks. | 0b0         |
| [0]     | HWEN059     | Enable use of PBHA[0] on memory accesses due to translation table walks using TTBR0_EL3. If this bit is clear, PBHA[0] will be 0 on translation table walks. | 0b0         |

## Access

MRS &lt;Xt&gt;, S3\_6\_C15\_C7\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0111 | 0b000 |

MSR S3\_6\_C15\_C7\_0, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0111 | 0b000 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C7\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_ATCR_EL3;
```

MSR S3\_6\_C15\_C7\_0, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    IMP_ATCR_EL3 = X[t, 64];
```

A.2.39 IMP\_CPUPSELR\_EL3, Selected Instruction Private Select Register

Selects the current instruction patch register for subsequent accesses to AArch64-IMP\_CPUPCR\_EL3, AArch64-IMP\_CPUPOR\_EL3, AArch64-IMP\_CPUPMR\_EL3, AArch64-IMP\_CPUPOR2\_EL3, AArch64-IMP\_CPUPMR2\_EL3, and AArch64-IMP\_CPUPFR\_EL3

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-45: AArch64\_imp\_cpupselr\_el3 bit assignments

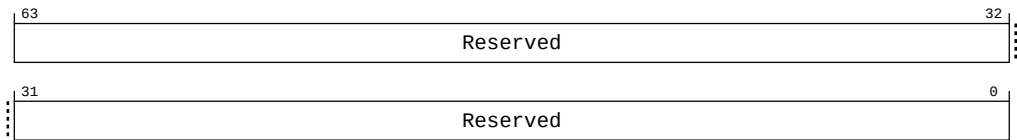


Table A-143: IMP\_CPUPSELR\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Access

MRS <Xt>, S3\_6\_C15\_C8\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b000 |

MSR S3\_6\_C15\_C8\_0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b000 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C8\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPSELR_EL3;
```

MSR S3\_6\_C15\_C8\_0, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
```



```
elseif PSTATE.EL == EL3 then
    IMP_CPUPSELR_EL3 = X[t, 64];
```

### A.2.40 IMP\_CPUPCR\_EL3, Selected Instruction Private Control Register

Configures current Instruction Patch selected by AArch64-IMP\_CPUPSELR\_EL3.SEL.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

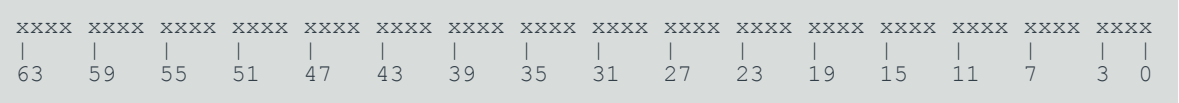
##### Functional group

Generic System Control

##### Access type

See bit descriptions

##### Reset value



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-46: AArch64\_imp\_cpupcr\_el3 bit assignments

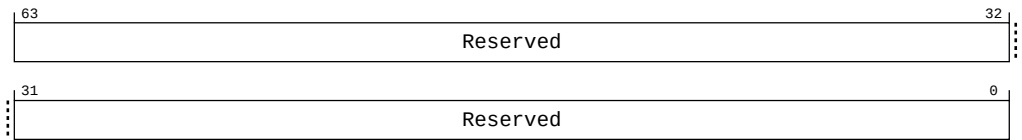


Table A-146: IMP\_CPUPCR\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

#### Access

MRS <Xt>, S3\_6\_C15\_C8\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b001 |

MSR S3\_6\_C15\_C8\_1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b001 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C8\_1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPCR_EL3;
```

MSR S3\_6\_C15\_C8\_1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    IMP_CPUPCR_EL3 = X[t, 64];
```

A.2.41 IMP\_CPUPOR\_EL3, Selected Instruction Private Opcode Register

Opcode for current Instruction Patch selected by AArch64-IMP\_CPUPSELR\_EL3.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

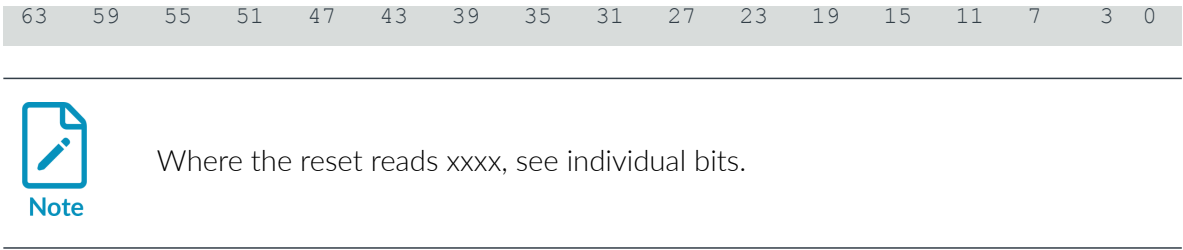
Generic System Control

Access type

See bit descriptions

Reset value





Bit descriptions

Figure A-47: AArch64\_imp\_cpupor\_el3 bit assignments

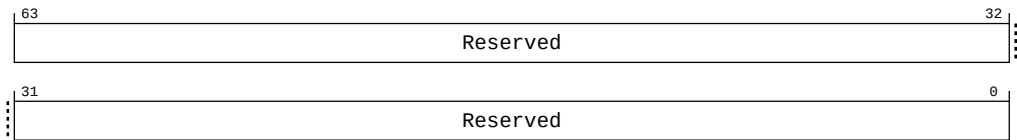


Table A-149: IMP\_CPUPOR\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_6\_C15\_C8\_2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b010 |

MSR S3\_6\_C15\_C8\_2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b010 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C8\_2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPOR_EL3;
```

MSR S3\_6\_C15\_C8\_2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
```

```
    UNDEFINED;
  elseif PSTATE.EL == EL2 then
    UNDEFINED;
  elseif PSTATE.EL == EL3 then
    IMP_CPUPOR_EL3 = X[t, 64];
```

A.2.42 IMP\_CPUPMR\_EL3, Selected Instruction Private Mask Register

Mask for current Instruction Patch selected by AArch64-IMP\_CPUPSELR\_EL3.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

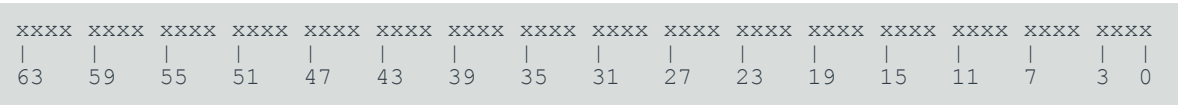
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-48: AArch64\_imp\_cpupmr\_el3 bit assignments



Table A-152: IMP\_CPUPMR\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

**Access**

MRS &lt;Xt&gt;, S3\_6\_C15\_C8\_3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b011 |

MSR S3\_6\_C15\_C8\_3, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b011 |

**Accessibility**

MRS &lt;Xt&gt;, S3\_6\_C15\_C8\_3

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPMR_EL3;

```

MSR S3\_6\_C15\_C8\_3, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    IMP_CPUPMR_EL3 = X[t, 64];

```

**A.2.43 IMP\_CPUPOR2\_EL3, Selected Instruction Private Opcode Register 2**

Opcode exclusion for current Instruction Patch selected by AArch64-IMP\_CPUPSELR\_EL3.SEL.

**Configurations**

This register is available in all configurations.

**Attributes****Width**

64

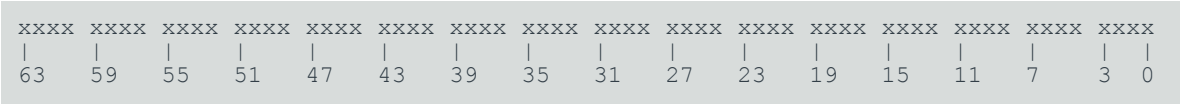
**Functional group**

Generic System Control

**Access type**

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-49: AArch64\_imp\_cpupor2\_el3 bit assignments

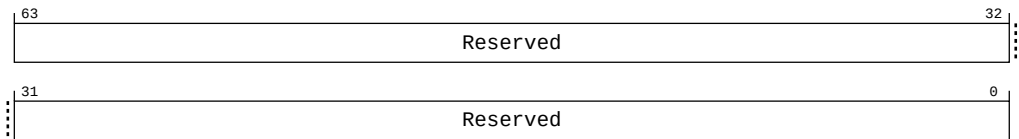


Table A-155: IMP\_CPUPOR2\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_6\_C15\_C8\_4

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b100 |

MSR S3\_6\_C15\_C8\_4, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b100 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C8\_4

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPOR2_EL3;
```

MSR S3\_6\_C15\_C8\_4, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    IMP_CPUPOR2_EL3 = X[t, 64];
```

A.2.44 IMP\_CPUPMR2\_EL3, Selected Instruction Private Mask Register 2

Mask exclusion for current Instruction Patch selected by AArch64-IMP\_CPUPSELR\_EL3.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Generic System Control

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-50: AArch64\_imp\_cpupmr2\_el3 bit assignments

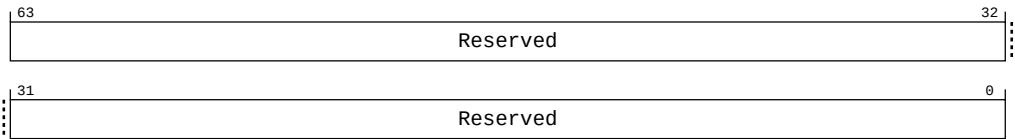


**Table A-158: IMP\_CPUPMR2\_EL3 bit descriptions**

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

**Access**

MRS &lt;Xt&gt;, S3\_6\_C15\_C8\_5

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b101 |

MSR S3\_6\_C15\_C8\_5, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b101 |

**Accessibility**

MRS &lt;Xt&gt;, S3\_6\_C15\_C8\_5

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPMR2_EL3;

```

MSR S3\_6\_C15\_C8\_5, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    IMP_CPUPMR2_EL3 = X[t, 64];

```

**A.2.45 IMP\_CPUPFR\_EL3, Selected Instruction Private Flag Register**

Instruction Patch flags for current Instruction Patch selected by AArch64-IMP\_CPUPSELR\_EL3.SEL.

**Configurations**

This register is available in all configurations.



Attributes

Width

64

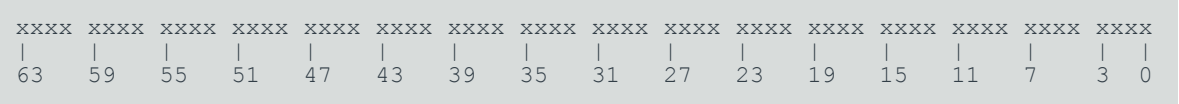
Functional group

Generic System Control

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-51: AArch64\_imp\_cpupfr\_el3 bit assignments

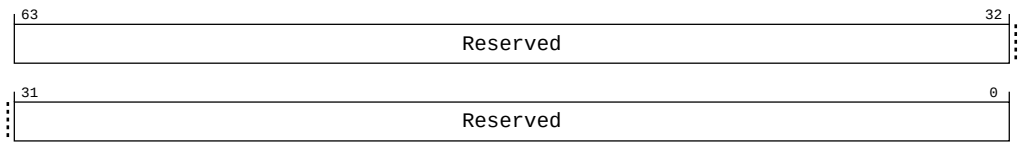


Table A-161: IMP\_CPUPFR\_EL3 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Access

MRS <Xt>, S3\_6\_C15\_C8\_6

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b110 |

MSR S3\_6\_C15\_C8\_6, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b1000 | 0b110 |

## Accessibility

MRS <Xt>, S3\_6\_C15\_C8\_6

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUPFR_EL3;

```

MSR S3\_6\_C15\_C8\_6, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    IMP_CPUPFR_EL3 = X[t, 64];

```

## A.3 AArch64 Debug registers summary

The following summary table provides an overview of all Debug registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-164: Debug registers summary**

| Name        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                             |
|-------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------|
| OSDTRRX_EL1 | 2   | 0   | C0  | C0  | 2   | See individual bit resets. | 64-bit | OS Lock Data Transfer Register, Receive |
| DBGBVR0_EL1 | 2   | 0   | C0  | C0  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers        |
| DBGBCR0_EL1 | 2   | 0   | C0  | C0  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers      |
| DBGWVR0_EL1 | 2   | 0   | C0  | C0  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers        |
| DBGWCR0_EL1 | 2   | 0   | C0  | C0  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers      |
| DBGBVR1_EL1 | 2   | 0   | C0  | C1  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers        |
| DBGBCR1_EL1 | 2   | 0   | C0  | C1  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers      |
| DBGWVR1_EL1 | 2   | 0   | C0  | C1  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers        |
| DBGWCR1_EL1 | 2   | 0   | C0  | C1  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers      |
| MDCCINT_EL1 | 2   | 0   | C0  | C2  | 0   | See individual bit resets. | 64-bit | Monitor DCC Interrupt Enable Register   |

| Name              | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|-------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| MDSCR_EL1         | 2   | 0   | C0  | C2  | 2   | See individual bit resets. | 64-bit | Monitor Debug System Control Register      |
| DBGBVR2_EL1       | 2   | 0   | C0  | C2  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR2_EL1       | 2   | 0   | C0  | C2  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| DBGWVR2_EL1       | 2   | 0   | C0  | C2  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers           |
| DBGWCR2_EL1       | 2   | 0   | C0  | C2  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers         |
| OSDTRTX_EL1       | 2   | 0   | C0  | C3  | 2   | See individual bit resets. | 64-bit | OS Lock Data Transfer Register, Transmit   |
| DBGBVR3_EL1       | 2   | 0   | C0  | C3  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR3_EL1       | 2   | 0   | C0  | C3  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| DBGWVR3_EL1       | 2   | 0   | C0  | C3  | 6   | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers           |
| DBGWCR3_EL1       | 2   | 0   | C0  | C3  | 7   | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers         |
| DBGBVR4_EL1       | 2   | 0   | C0  | C4  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR4_EL1       | 2   | 0   | C0  | C4  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| DBGBVR5_EL1       | 2   | 0   | C0  | C5  | 4   | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers           |
| DBGBCR5_EL1       | 2   | 0   | C0  | C5  | 5   | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers         |
| OSECCR_EL1        | 2   | 0   | C0  | C6  | 2   | See individual bit resets. | 64-bit | OS Lock Exception Catch Control Register   |
| MDRAR_EL1         | 2   | 0   | C1  | C0  | 0   | See individual bit resets. | 64-bit | Monitor Debug ROM Address Register         |
| OSLAR_EL1         | 2   | 0   | C1  | C0  | 4   | See individual bit resets. | 64-bit | OS Lock Access Register                    |
| OSLSR_EL1         | 2   | 0   | C1  | C1  | 4   | See individual bit resets. | 64-bit | OS Lock Status Register                    |
| OSDLR_EL1         | 2   | 0   | C1  | C3  | 4   | See individual bit resets. | 64-bit | OS Double Lock Register                    |
| DBGPRCR_EL1       | 2   | 0   | C1  | C4  | 4   | See individual bit resets. | 64-bit | Debug Power Control Register               |
| DBGCLAIMSET_EL1   | 2   | 0   | C7  | C8  | 6   | See individual bit resets. | 64-bit | Debug CLAIM Tag Set Register               |
| DBGCLAIMCLR_EL1   | 2   | 0   | C7  | C9  | 6   | See individual bit resets. | 64-bit | Debug CLAIM Tag Clear Register             |
| DBGAUTHSTATUS_EL1 | 2   | 0   | C7  | C14 | 6   | See individual bit resets. | 64-bit | Debug Authentication Status Register       |
| MDCCSR_ELO        | 2   | 3   | C0  | C1  | 0   | See individual bit resets. | 64-bit | Monitor DCC Status Register                |
| DBGDTR_ELO        | 2   | 3   | C0  | C4  | 0   | See individual bit resets. | 64-bit | Debug Data Transfer Register, half-duplex  |
| DBGDTRRX_ELO      | 2   | 3   | C0  | C5  | 0   | See individual bit resets. | 64-bit | Debug Data Transfer Register, Receive      |
| DBGDTRTX_ELO      | 2   | 3   | C0  | C5  | 0   | See individual bit resets. | 64-bit | Debug Data Transfer Register, Transmit     |
| TRFCR_EL1         | 3   | 0   | C1  | C2  | 1   | See individual bit resets. | 64-bit | Trace Filter Control Register (EL1)        |
| MDCR_EL2          | 3   | 4   | C1  | C1  | 1   | See individual bit resets. | 64-bit | Monitor Debug Configuration Register (EL2) |
| TRFCR_EL2         | 3   | 4   | C1  | C2  | 1   | See individual bit resets. | 64-bit | Trace Filter Control Register (EL2)        |
| IMP_IDATA0_EL3    | 3   | 6   | C15 | C0  | 0   | See individual bit resets. | 64-bit | Instruction Register 0                     |
| IMP_IDATA1_EL3    | 3   | 6   | C15 | C0  | 1   | See individual bit resets. | 64-bit | Instruction Register 1                     |
| IMP_IDATA2_EL3    | 3   | 6   | C15 | C0  | 2   | See individual bit resets. | 64-bit | Instruction Register 2                     |
| IMP_DDATA0_EL3    | 3   | 6   | C15 | C1  | 0   | See individual bit resets. | 64-bit | Data Register 0                            |
| IMP_DDATA1_EL3    | 3   | 6   | C15 | C1  | 1   | See individual bit resets. | 64-bit | Data Register 1                            |
| IMP_DDATA2_EL3    | 3   | 6   | C15 | C1  | 2   | See individual bit resets. | 64-bit | Data Register 2                            |

### A.3.1 IMP\_IDATA0\_EL3, Instruction Register 0

Contains data from a preceeding RAMINDEX operation.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

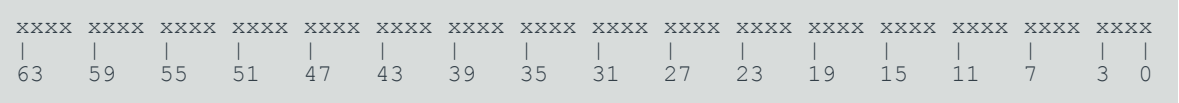
##### Functional group

Debug registers

##### Access type

See bit descriptions

##### Reset value



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-52: AArch64\_imp\_idata0\_el3 bit assignments

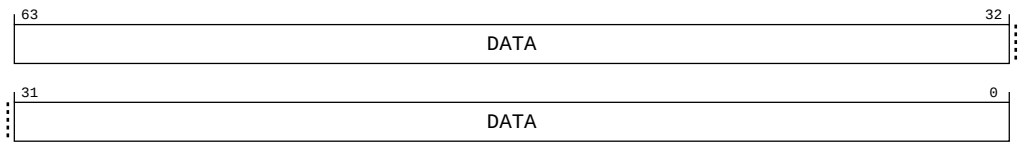


Table A-165: IMP\_IDATA0\_EL3 bit descriptions

| Bits   | Name | Description                                       | Reset    |
|--------|------|---------------------------------------------------|----------|
| [63:0] | DATA | Contains data from a preceding RAMINDEX operation | 64 { x } |

#### Access

MRS <Xt>, S3\_6\_C15\_C0\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0000 | 0b000 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C0\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_IDATA0_EL3;
```

A.3.2 IMP\_IDATA1\_EL3, Instruction Register 1

Contains data from a preceeding RAMINDEX operation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Debug registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-53: AArch64\_imp\_idata1\_el3 bit assignments

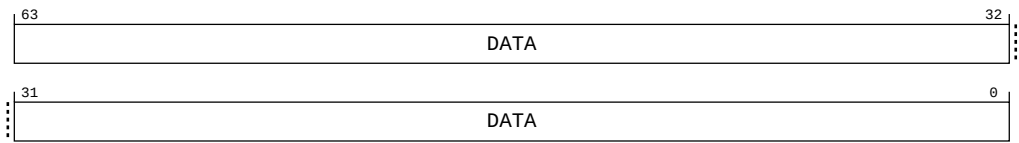


Table A-167: IMP\_IDATA1\_EL3 bit descriptions

| Bits   | Name | Description                                       | Reset  |
|--------|------|---------------------------------------------------|--------|
| [63:0] | DATA | Contains data from a preceding RAMINDEX operation | 64 {x} |

**Access**  
MRS <Xt>, S3\_6\_C15\_CO\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0000 | 0b001 |

**Accessibility**  
MRS <Xt>, S3\_6\_C15\_CO\_1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_IDATA1_EL3;
```

A.3.3 IMP\_IDATA2\_EL3, Instruction Register 2

Contains data from a preceeding RAMINDEX operation.

**Configurations**  
This register is available in all configurations.

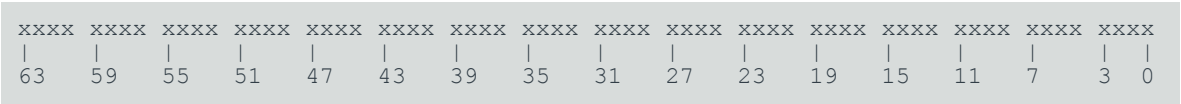
**Attributes**

**Width**  
64

**Functional group**  
Debug registers

**Access type**  
See bit descriptions

**Reset value**





Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-54: AArch64\_imp\_idata2\_el3 bit assignments

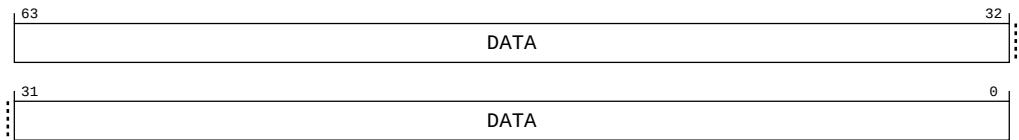


Table A-169: IMP\_IDATA2\_EL3 bit descriptions

| Bits   | Name | Description                                       | Reset    |
|--------|------|---------------------------------------------------|----------|
| [63:0] | DATA | Contains data from a preceding RAMINDEX operation | 64 { x } |

Access

MRS <Xt>, S3\_6\_C15\_C0\_2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0000 | 0b010 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C0\_2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = IMP_IDATA2_EL3;
```

A.3.4 IMP\_DDATA0\_EL3, Data Register 0

Contains data from a preceeding RAMINDEX operation.

Configurations

This register is available in all configurations.

Attributes

Width

64

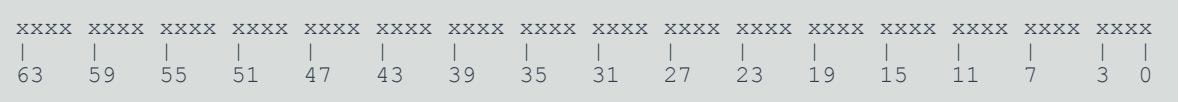
Functional group

Debug registers

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-55: AArch64\_imp\_ddata0\_el3 bit assignments

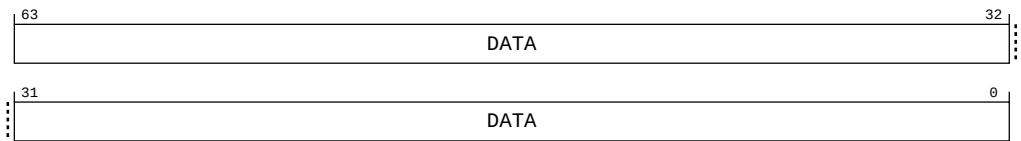


Table A-171: IMP\_DDATA0\_EL3 bit descriptions

| Bits   | Name | Description                                       | Reset  |
|--------|------|---------------------------------------------------|--------|
| [63:0] | DATA | Contains data from a preceding RAMINDEX operation | 64 {x} |

Access

MRS <Xt>, S3\_6\_C15\_C1\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0001 | 0b000 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C1\_0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
```



```
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_DDATA0_EL3;
```

A.3.5 IMP\_DDATA1\_EL3, Data Register 1

Contains data from a preceeding RAMINDEX operation.

Configurations

This register is available in all configurations.

Attributes

Width

64

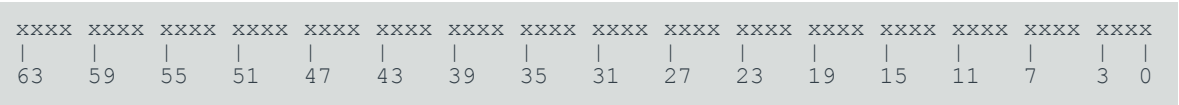
Functional group

Debug registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-56: AArch64\_imp\_ddata1\_el3 bit assignments

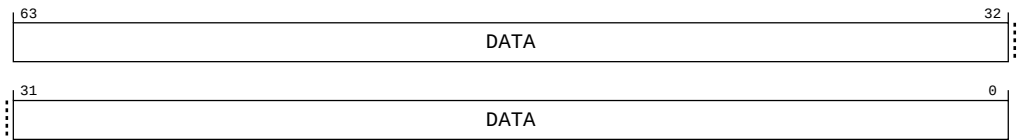


Table A-173: IMP\_DDATA1\_EL3 bit descriptions

| Bits   | Name | Description                                       | Reset  |
|--------|------|---------------------------------------------------|--------|
| [63:0] | DATA | Contains data from a preceding RAMINDEX operation | 64 {x} |

Access

MRS <Xt>, S3\_6\_C15\_C1\_1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0001 | 0b001 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C1\_1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_DDATA1_EL3;
```

A.3.6 IMP\_DDATA2\_EL3, Data Register 2

Contains data from a preceeding RAMINDEX operation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Debug registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-57: AArch64\_imp\_ddata2\_el3 bit assignments

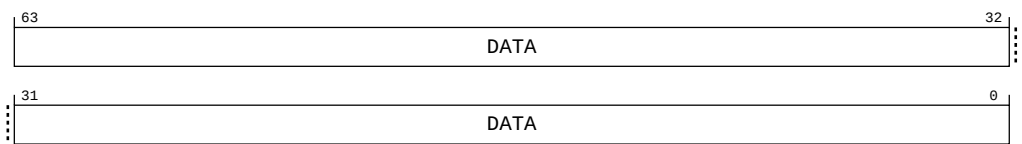


Table A-175: IMP\_DDATA2\_EL3 bit descriptions

| Bits   | Name | Description                                       | Reset  |
|--------|------|---------------------------------------------------|--------|
| [63:0] | DATA | Contains data from a preceding RAMINDEX operation | 64 {x} |

Access

MRS <Xt>, S3\_6\_C15\_C1\_2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1111 | 0b0001 | 0b010 |

Accessibility

MRS <Xt>, S3\_6\_C15\_C1\_2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_DDATA2_EL3;
```

A.4 AArch64 System instructions summary

The following summary table provides an overview of all System instructions in the core.

For more information on registers listed in the table, click on the link associated with the register name.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

Table A-177: System instructions summary

| Name                             | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description |
|----------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------|
| <a href="#">SYS_IMP_RAMINDEX</a> | 1   | 6   | C15 | C0  | 0   | See individual bit resets. | 64-bit | RAM Index   |

### A.4.1 SYS\_IMP\_RAMINDEX, RAM Index

Read contents of the cache specified by the source register into AArch64-IMP\_IDATA0\_EL3, AArch64-IMP\_IDATA1\_EL3, AArch64-IMP\_IDATA2\_EL3, AArch64-IMP\_DDATA0\_EL3, AArch64-IMP\_DDATA1\_EL3, and AArch64-IMP\_DDATA2\_EL3.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

System instructions

##### Access type

See bit descriptions

#### Bit descriptions

Figure A-58: AArch64\_sys\_imp\_ramindex bit assignments

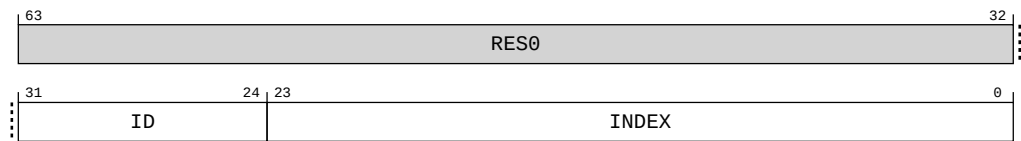


Table A-178: SYS\_IMP\_RAMINDEX bit descriptions

| Bits    | Name  | Description                | Reset  |
|---------|-------|----------------------------|--------|
| [63:32] | RES0  | Reserved                   | RES0   |
| [31:24] | ID    | RAM ID (See Chapter 10)    | 8 {x}  |
| [23:0]  | INDEX | RAM Index (See Chapter 10) | 24 {x} |

#### Access

SYS #6, C15, C0, #0{, <Xt>}

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b01 | 0b110 | 0b1111 | 0b0000 | 0b000 |

#### Accessibility

SYS #6, C15, C0, #0{, <Xt>}

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
```

```

elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    SYS_IMP_RAMINDEX(X[t, 64]);

```

## A.5 AArch64 Identification registers summary

The following summary table provides an overview of all Identification registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-180: Identification registers summary**

| Name                       | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                  |
|----------------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------|
| <a href="#">MIDR_EL1</a>   | 3   | 0   | C0  | C0  | 0   | See individual bit resets. | 64-bit | Main ID Register                             |
| <a href="#">MPIDR_EL1</a>  | 3   | 0   | C0  | C0  | 5   | See individual bit resets. | 64-bit | Multiprocessor Affinity Register             |
| <a href="#">REVIDR_EL1</a> | 3   | 0   | C0  | C0  | 6   | See individual bit resets. | 64-bit | Revision ID Register                         |
| ID_PFR0_EL1                | 3   | 0   | C0  | C1  | 0   | See individual bit resets. | 64-bit | AArch32 Processor Feature Register 0         |
| ID_PFR1_EL1                | 3   | 0   | C0  | C1  | 1   | See individual bit resets. | 64-bit | AArch32 Processor Feature Register 1         |
| ID_DFR0_EL1                | 3   | 0   | C0  | C1  | 2   | See individual bit resets. | 64-bit | AArch32 Debug Feature Register 0             |
| ID_AFR0_EL1                | 3   | 0   | C0  | C1  | 3   | See individual bit resets. | 64-bit | AArch32 Auxiliary Feature Register 0         |
| ID_MMFR0_EL1               | 3   | 0   | C0  | C1  | 4   | See individual bit resets. | 64-bit | AArch32 Memory Model Feature Register 0      |
| ID_MMFR1_EL1               | 3   | 0   | C0  | C1  | 5   | See individual bit resets. | 64-bit | AArch32 Memory Model Feature Register 1      |
| ID_MMFR2_EL1               | 3   | 0   | C0  | C1  | 6   | See individual bit resets. | 64-bit | AArch32 Memory Model Feature Register 2      |
| ID_MMFR3_EL1               | 3   | 0   | C0  | C1  | 7   | See individual bit resets. | 64-bit | AArch32 Memory Model Feature Register 3      |
| ID_ISAR0_EL1               | 3   | 0   | C0  | C2  | 0   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 0 |
| ID_ISAR1_EL1               | 3   | 0   | C0  | C2  | 1   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 1 |
| ID_ISAR2_EL1               | 3   | 0   | C0  | C2  | 2   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 2 |
| ID_ISAR3_EL1               | 3   | 0   | C0  | C2  | 3   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 3 |
| ID_ISAR4_EL1               | 3   | 0   | C0  | C2  | 4   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 4 |
| ID_ISAR5_EL1               | 3   | 0   | C0  | C2  | 5   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 5 |
| ID_MMFR4_EL1               | 3   | 0   | C0  | C2  | 6   | See individual bit resets. | 64-bit | AArch32 Memory Model Feature Register 4      |
| ID_ISAR6_EL1               | 3   | 0   | C0  | C2  | 7   | See individual bit resets. | 64-bit | AArch32 Instruction Set Attribute Register 6 |
| MVFR0_EL1                  | 3   | 0   | C0  | C3  | 0   | See individual bit resets. | 64-bit | AArch32 Media and VFP Feature Register 0     |
| MVFR1_EL1                  | 3   | 0   | C0  | C3  | 1   | See individual bit resets. | 64-bit | AArch32 Media and VFP Feature Register 1     |
| MVFR2_EL1                  | 3   | 0   | C0  | C3  | 2   | See individual bit resets. | 64-bit | AArch32 Media and VFP Feature Register 2     |
| ID_PFR2_EL1                | 3   | 0   | C0  | C3  | 4   | See individual bit resets. | 64-bit | AArch32 Processor Feature Register 2         |

| Name             | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                  |
|------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------|
| ID_DFR1_EL1      | 3   | 0   | C0  | C3  | 5   | See individual bit resets. | 64-bit | Debug Feature Register 1                     |
| ID_MMFR5_EL1     | 3   | 0   | C0  | C3  | 6   | See individual bit resets. | 64-bit | AArch32 Memory Model Feature Register 5      |
| ID_AA64PFR0_EL1  | 3   | 0   | C0  | C4  | 0   | See individual bit resets. | 64-bit | AArch64 Processor Feature Register 0         |
| ID_AA64PFR1_EL1  | 3   | 0   | C0  | C4  | 1   | See individual bit resets. | 64-bit | AArch64 Processor Feature Register 1         |
| ID_AA64ZFR0_EL1  | 3   | 0   | C0  | C4  | 4   | See individual bit resets. | 64-bit | SVE Feature ID Register 0                    |
| ID_AA64DFR0_EL1  | 3   | 0   | C0  | C5  | 0   | See individual bit resets. | 64-bit | AArch64 Debug Feature Register 0             |
| ID_AA64DFR1_EL1  | 3   | 0   | C0  | C5  | 1   | See individual bit resets. | 64-bit | AArch64 Debug Feature Register 1             |
| ID_AA64AFR0_EL1  | 3   | 0   | C0  | C5  | 4   | See individual bit resets. | 64-bit | AArch64 Auxiliary Feature Register 0         |
| ID_AA64AFR1_EL1  | 3   | 0   | C0  | C5  | 5   | See individual bit resets. | 64-bit | AArch64 Auxiliary Feature Register 1         |
| ID_AA64ISAR0_EL1 | 3   | 0   | C0  | C6  | 0   | See individual bit resets. | 64-bit | AArch64 Instruction Set Attribute Register 0 |
| ID_AA64ISAR1_EL1 | 3   | 0   | C0  | C6  | 1   | See individual bit resets. | 64-bit | AArch64 Instruction Set Attribute Register 1 |
| ID_AA64ISAR2_EL1 | 3   | 0   | C0  | C6  | 2   | See individual bit resets. | 64-bit | AArch64 Instruction Set Attribute Register 2 |
| ID_AA64MMFR0_EL1 | 3   | 0   | C0  | C7  | 0   | See individual bit resets. | 64-bit | AArch64 Memory Model Feature Register 0      |
| ID_AA64MMFR1_EL1 | 3   | 0   | C0  | C7  | 1   | See individual bit resets. | 64-bit | AArch64 Memory Model Feature Register 1      |
| ID_AA64MMFR2_EL1 | 3   | 0   | C0  | C7  | 2   | See individual bit resets. | 64-bit | AArch64 Memory Model Feature Register 2      |
| ID_AA64MMFR3_EL1 | 3   | 0   | C0  | C7  | 3   | See individual bit resets. | 64-bit | AArch64 Memory Model Feature Register 3      |
| MPAMIDR_EL1      | 3   | 0   | C10 | C4  | 4   | See individual bit resets. | 64-bit | MPAM ID Register (EL1)                       |
| IMP_CPUCFR_EL1   | 3   | 0   | C15 | C0  | 0   | See individual bit resets. | 64-bit | CPU Configuration Register                   |
| CCSIDR_EL1       | 3   | 1   | C0  | C0  | 0   | See individual bit resets. | 64-bit | Current Cache Size ID Register               |
| CLIDR_EL1        | 3   | 1   | C0  | C0  | 1   | See individual bit resets. | 64-bit | Cache Level ID Register                      |
| CCSIDR2_EL1      | 3   | 1   | C0  | C0  | 2   | See individual bit resets. | 64-bit | Current Cache Size ID Register 2             |
| GMID_EL1         | 3   | 1   | C0  | C0  | 4   | See individual bit resets. | 64-bit | Multiple tag transfer ID Register            |
| CSSELR_EL1       | 3   | 2   | C0  | C0  | 0   | See individual bit resets. | 64-bit | Cache Size Selection Register                |
| CTR_EL0          | 3   | 3   | C0  | C0  | 1   | See individual bit resets. | 64-bit | Cache Type Register                          |
| DCZID_EL0        | 3   | 3   | C0  | C0  | 7   | See individual bit resets. | 64-bit | Data Cache Zero ID Register                  |
| VPIDR_EL2        | 3   | 4   | C0  | C0  | 0   | See individual bit resets. | 64-bit | Virtualization Processor ID Register         |
| VMPIDR_EL2       | 3   | 4   | C0  | C0  | 5   | See individual bit resets. | 64-bit | Virtualization Multiprocessor ID Register    |

### A.5.1 MIDR\_EL1, Main ID Register

Provides identification information for the PE, including an implementer code for the device and a device ID number.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

Functional group  
Identification registers

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0100 | 0001 | 0000 | 1111 | 1101 | 1000 | 0101 | 0010 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-59: AArch64\_midr\_el1 bit assignments

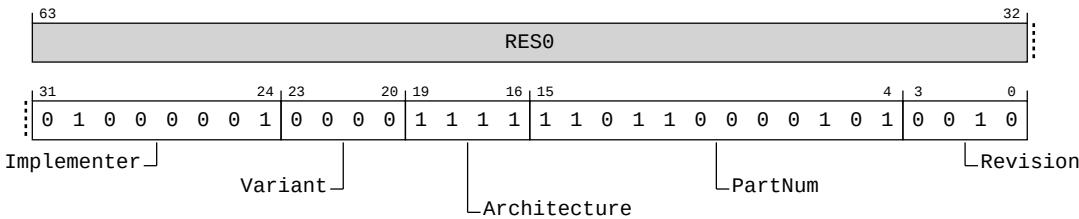


Table A-181: MIDR\_EL1 bit descriptions

| Bits    | Name         | Description                                                                                                                                                                                                                                    | Reset  |
|---------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:32] | RES0         | Reserved                                                                                                                                                                                                                                       | RES0   |
| [31:24] | Implementer  | Indicates the implementer code. This value is:<br><br><b>0b01000001</b><br>Arm Limited.                                                                                                                                                        | 0x41   |
| [23:20] | Variant      | Variant number. Typically, this field is used to distinguish between different product variants, or major revisions of a product.<br><br><b>0b0000</b><br>r0p2                                                                                 | 0b0000 |
| [19:16] | Architecture | Indicates the architecture code. This value is:<br><br><b>0b1111</b><br>Architecture is defined by ID registers                                                                                                                                | 0b1111 |
| [15:4]  | PartNum      | Primary Part Number for the device.<br><br>On processors implemented by Arm, if the top four bits of the primary part number are 0x0 or 0x7, the variant and architecture are encoded differently.<br><br><b>0b110110000101</b><br>Cortex-X925 | 0xD85  |

| Bits  | Name     | Description                                                  | Reset  |
|-------|----------|--------------------------------------------------------------|--------|
| [3:0] | Revision | Revision number for the device.<br><br><b>0b0010</b><br>rOp2 | 0b0010 |

### Access

MRS &lt;Xt&gt;, MIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0000 | 0b000 |

### Accessibility

MRS &lt;Xt&gt;, MIDR\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.MIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() then
        X[t, 64] = VPIDR_EL2;
    else
        X[t, 64] = MIDR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = MIDR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = MIDR_EL1;

```

## A.5.2 MPIDR\_EL1, Multiprocessor Affinity Register

In a multiprocessor system, provides an additional PE identification mechanism for scheduling purposes.

### Configurations

In a uniprocessor system, Arm recommends that each Aff<n> field of this register returns a value of 0.

### Attributes

#### Width

64

#### Functional group

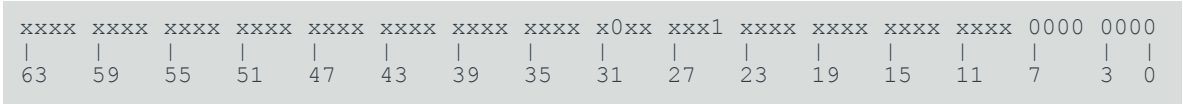
Identification registers

#### Access type

See bit descriptions



Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-60: AArch64\_mpidr\_el1 bit assignments

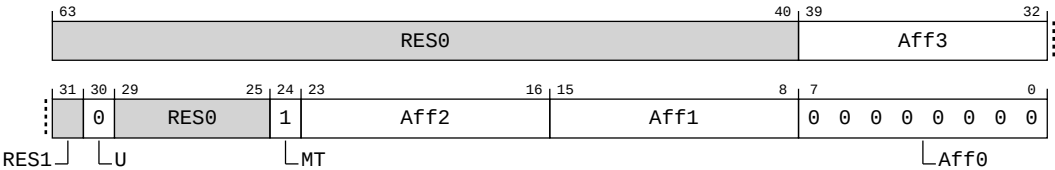


Table A-183: MPIDR\_EL1 bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                     | Reset |
|---------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:40] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                        | RES0  |
| [39:32] | Aff3 | Affinity level 3. See the description of Aff0 for more information.<br><br>The value will be determined by the CLUSTERIDAFF3 configuration pins.                                                                                                                                                                                                                | 8 {x} |
| [31]    | RES1 | Reserved                                                                                                                                                                                                                                                                                                                                                        | RES1  |
| [30]    | U    | Indicates a Uniprocessor system, as distinct from PE 0 in a multiprocessor system.<br><br>0b0<br>Processor is part of a multiprocessor system.                                                                                                                                                                                                                  | 0b0   |
| [29:25] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                        | RES0  |
| [24]    | MT   | Indicates whether the lowest level of affinity consists of logical PEs that are implemented using a multithreading type approach. See the description of Aff0 for more information about affinity levels.<br><br>0b1<br>Performance of PEs with different affinity level 0 values, and the same values for affinity level 1 and higher, is very interdependent. | 0b1   |
| [23:16] | Aff2 | Affinity level 2. See the description of Aff0 for more information.<br><br>The value will be determined by the CLUSTERIDAFF2 configuration pins.                                                                                                                                                                                                                | 8 {x} |
| [15:8]  | Aff1 | Affinity level 1. See the description of Aff0 for more information.<br><br>Value read from the CPUID configuration pins. Identification number for each CPU in an cluster counting from zero.                                                                                                                                                                   | 8 {x} |

| Bits  | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7:0] | Aff0 | Affinity level 0. This is the affinity level that is most significant for determining PE behavior. Higher affinity levels are increasingly less significant in determining PE behavior. The assigned value of the MPIDR.{Aff2, Aff1, Aff0} or AArch64-MPIDR_EL1.{Aff3, Aff2, Aff1, Aff0} set of fields of each PE must be unique within the system as a whole.<br><br><b>0b00000000</b><br>Only one thread. | 0x00  |

### Access

MRS <Xt>, MPIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0000 | 0b101 |

### Accessibility

MRS <Xt>, MPIDR\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.MPIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() then
        X[t, 64] = VMPIDR_EL2;
    else
        X[t, 64] = MPIDR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = MPIDR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = MPIDR_EL1;

```

## A.5.3 REVIDR\_EL1, Revision ID Register

The REVIDR\_EL1 provides revision information, additional to MIDR\_EL1, that identifies minor fixes (errata) which might be present in a specific implementation of the Cortex-X925 core. Refer to the Cortex-X925 Product Errata Notice (PEN) for information on how to interpret the values in this register.

### Configurations

If REVIDR\_EL1 has the same value as AArch64-MIDR\_EL1, then its contents have no significance.

### Attributes

#### Width

64

#### Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-61: AArch64\_revidr\_el1 bit assignments

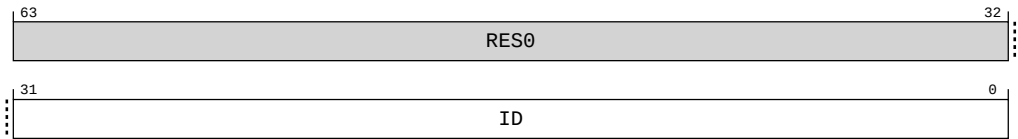


Table A-185: REVIDR\_EL1 bit descriptions

| Bits    | Name | Description | Reset  |
|---------|------|-------------|--------|
| [63:32] | RES0 | Reserved    | RES0   |
| [31:0]  | ID   | None        | 32 {x} |

Access

MRS <Xt>, REVIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0000 | 0b110 |

Accessibility

MRS <Xt>, REVIDR\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.REVIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = REVIDR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = REVIDR_EL1;
```

```
elseif PSTATE.EL == EL3 then
    X[t, 64] = REVIDR_EL1;
```

### A.5.4 ID\_AA64PFR0\_EL1, AArch64 Processor Feature Register 0

Provides additional information about implemented PE features in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

The external register ext-EDPFR gives information from this register.

#### Attributes

##### Width

64

##### Functional group

Identification registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0001 | 0010 | 0000 | 0001 | 0001 | 0001 | 0001 | 0001 | 0010 | xxxx | 0001 | 0001 | 0001 | 0001 | 0001 | 0001 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-62: AArch64\_id\_aa64pfr0\_el1 bit assignments

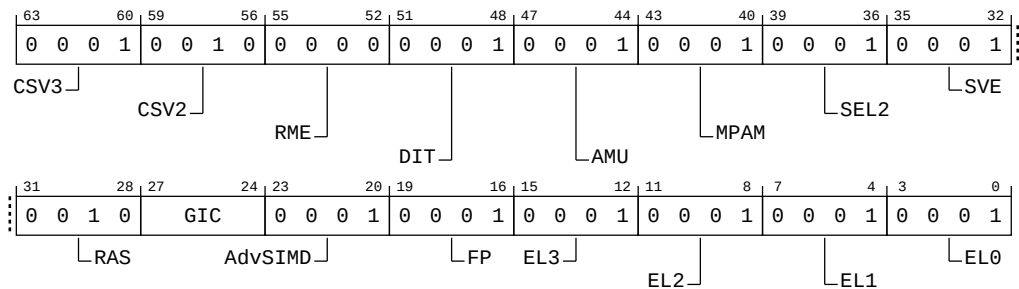


**Table A-187: ID\_AA64PFR0\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset                                                                                 |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| [63:60] | CSV3 | Speculative use of faulting data. Defined values are:<br><b>0b0001</b><br>Data loaded under speculation with a permission or domain fault cannot be used to form an address, generate condition codes, or generate SVE predicate values to be used by other instructions in the speculative sequence. The execution timing of any other instructions in the speculative sequence is not a function of the data loaded under speculation. | 0b0001                                                                                |
| [59:56] | CSV2 | Speculative use of out of context branch targets. Defined values are:<br><b>0b0010</b><br>FEAT_CSV2_2 is implemented, but FEAT_CSV2_3 is not implemented.                                                                                                                                                                                                                                                                                | 0b0010                                                                                |
| [55:52] | RME  | Realm Management Extension (RME). Defined values are:<br><b>0b0000</b><br>Realm Management Extension not implemented.                                                                                                                                                                                                                                                                                                                    | 0b0000                                                                                |
| [51:48] | DIT  | Data Independent Timing. Defined values are:<br><b>0b0001</b><br>AArch64 provides the PSTATE.DIT mechanism to guarantee constant execution time of certain instructions.                                                                                                                                                                                                                                                                 | 0b0001                                                                                |
| [47:44] | AMU  | Indicates support for Activity Monitors Extension. Defined values are:<br><b>0b0001</b><br>FEAT_AMUv1 is implemented.                                                                                                                                                                                                                                                                                                                    | 0b0001                                                                                |
| [43:40] | MPAM | Indicates the major version number of support for the MPAM Extension.<br><br>Defined values are:<br><b>0b0001</b><br>The major version number of the MPAM extension is 1.                                                                                                                                                                                                                                                                | 0b0001                                                                                |
| [39:36] | SEL2 | Secure EL2. Defined values are:<br><b>0b0001</b><br>Secure EL2 is implemented.                                                                                                                                                                                                                                                                                                                                                           | 0b0001                                                                                |
| [35:32] | SVE  | Scalable Vector Extension. Defined values are:<br><b>0b0001</b><br>SVE architectural state and programmers' model are implemented.                                                                                                                                                                                                                                                                                                       | 0b0001                                                                                |
| [31:28] | RAS  | RAS Extension version. Defined values are:<br><b>0b0010</b><br>ARMv8.4-RAS present. As 0b0001, and adds support for ARMv8.4-DFE (If EL3 is implemented), additional ERXMISCM_EL1 System registers, additionalSystem registers ERXPFGCDN_EL1, ERXPFGCTL_EL1, and ERXPFGF_EL1, and the SCR_EL3.FIEN and HCR_EL2.FIEN trap controls, to support the optional RAS Common Fault Injection Model Extension.                                    | 0b0010                                                                                |
| [27:24] | GIC  | System register GIC CPU interface. Defined values are:<br><b>0b0000</b><br>When Port GICCDISABLE is High, GIC CPU interface is disabled.<br><b>0b0011</b><br>When Port GICCDISABLE is Low, GIC (version 4.1) CPU interface is enabled.                                                                                                                                                                                                   | The reset values can be the following:<br>0b0000, 0b0011,<br>respective to the value. |

| Bits    | Name    | Description                                                                                                                                                 | Reset  |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [23:20] | AdvSIMD | Advanced SIMD. Defined values are:<br><br><b>0b0001</b><br>Advanced SIMD is implemented, including support for half-precision floating-point arithmetic.    | 0b0001 |
| [19:16] | FP      | Floating-point. Defined values are:<br><br><b>0b0001</b><br>Floating-point, including support for half-precision floating-point arithmetic, is implemented. | 0b0001 |
| [15:12] | EL3     | EL3 Exception level handling. Defined values are:<br><br><b>0b0001</b><br>EL3 can be executed in AArch64 state only.                                        | 0b0001 |
| [11:8]  | EL2     | EL2 Exception level handling. Defined values are:<br><br><b>0b0001</b><br>EL2 can be executed in AArch64 state only.                                        | 0b0001 |
| [7:4]   | EL1     | EL1 Exception level handling. Defined values are:<br><br><b>0b0001</b><br>EL1 can be executed in AArch64 state only.                                        | 0b0001 |
| [3:0]   | ELO     | ELO Exception level handling. Defined values are:<br><br><b>0b0001</b><br>ELO can be executed in AArch64 state only.                                        | 0b0001 |

## Access

MRS <Xt>, ID\_AA64PFR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0100 | 0b000 |

## Accessibility

MRS <Xt>, ID\_AA64PFR0\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64PFR0_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64PFR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64PFR0_EL1;

```

### A.5.5 ID\_AA64PFR1\_EL1, AArch64 Processor Feature Register 1

Reserved for future expansion of information about implemented PE features in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Identification registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | xxxx | 0001 | xxxx | xxxx | 0010 | 0001 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-63: AArch64\_id\_aa64pfr1\_el1 bit assignments

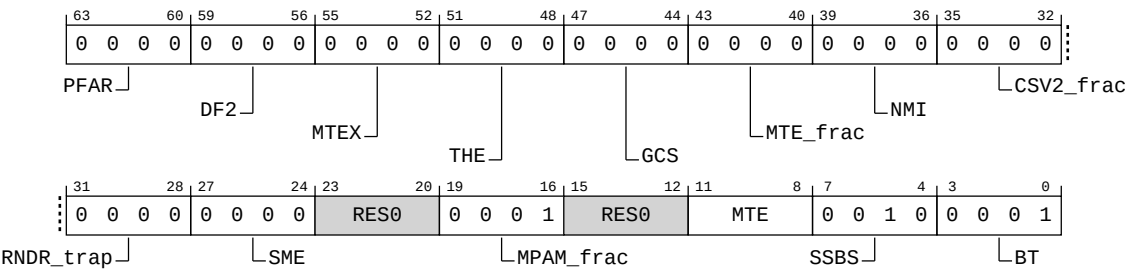


**Table A-189: ID\_AA64PFR1\_EL1 bit descriptions**

| Bits    | Name      | Description                                                                                                                                                                                                                                                                                   | Reset  |
|---------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | PFAR      | Support for physical fault address registers, FEAT_P FAR. Defined values are:<br><b>0b0000</b><br>FEAT_P FAR is not implemented.                                                                                                                                                              | 0b0000 |
| [59:56] | DF2       | Support for error exception routing extensions, FEAT_DoubleFault2. Defined values are:<br><b>0b0000</b><br>FEAT_DoubleFault2 is not implemented.<br><br><b>Note:</b><br>This does not mean that FEAT_DoubleFault, as identified by AArch64-ID_AA64PFR0_EL1.RAS >= 0b0010, is not implemented. | 0b0000 |
| [55:52] | MTEX      | Additional tag checking modes for MTE. Defined values are:<br><b>0b0000</b><br>Support for Memory Tagging when Address tagging is enabled.                                                                                                                                                    | 0b0000 |
| [51:48] | THE       | Support for Translation Hardening Extension. Defined values are:<br><b>0b0000</b><br>Translation Hardening Extension is not implemented.                                                                                                                                                      | 0b0000 |
| [47:44] | GCS       | Support for Guarded Control Stack. Defined values are:<br><b>0b0000</b><br>Guarded Control Stack is not implemented.                                                                                                                                                                          | 0b0000 |
| [43:40] | MTE_frac  | Support for Asynchronous Faulting and asymmetric Tag Check Fault handling. Defined values are:<br><b>0b0000</b><br>Asynchronous Faulting is supported.<br><br>If ID_AA64PFR1_EL1.MTE >= 0b0011, asymmetric Tag Check Fault handling is supported.                                             | 0b0000 |
| [39:36] | NMI       | Non-maskable Interrupt. Indicates support for Non-maskable interrupts. Defined values are:<br><b>0b0000</b><br>SCTLR_ELx.{SPINTMASK, NMI} and PSTATE.ALLINT with its associated instructions are not supported.                                                                               | 0b0000 |
| [35:32] | CSV2_frac | CSV2 fractional field. Defined values are:<br><b>0b0000</b><br>Either AArch64-ID_AA64PFR0_EL1.CSV2 is not 0b0001, or the implementation does not disclose whether FEAT_CSV2_1p1 is implemented.<br><br>FEAT_CSV2_1p2 is not implemented.                                                      | 0b0000 |
| [31:28] | RNDR_trap | Random Number trap to EL3 field. Defined values are:<br><b>0b0000</b><br>Trapping of AArch64-RNDR and AArch64-RNDRS to EL3 is not supported.                                                                                                                                                  | 0b0000 |
| [27:24] | SME       | Scalable Matrix Extension. Defined values are:<br><b>0b0000</b><br>SME architectural state and programmers' model are not implemented.                                                                                                                                                        | 0b0000 |



| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset                                                                                   |
|---------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| [23:20] | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>RES0</b>                                                                             |
| [19:16] | MPAM_frac   | Indicates the minor version number of support for the MPAM Extension.<br><br>Defined values are:<br><br><b>0b0001</b><br>The minor version number of the MPAM extension is 1.                                                                                                                                                                                                                                                                                                                                                                                                                         | 0b0001                                                                                  |
| [15:12] | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>RES0</b>                                                                             |
| [11:8]  | MTE         | Support for the Memory Tagging Extension. Defined values are:<br><br><b>0b0001</b><br>Memory Tagging Extension instructions accessible at EL0 are implemented. Instructions and System Registers defined by the extension not configurably accessible at EL0 are Unallocated and other System Register fields defined by the extension are <b>RES0</b> . This value is reported when the BROADCASTMTE input is LOW.<br><br><b>0b0011</b><br>Memory Tagging Extension is implemented with support for asymmetric Tag Check Fault handling. This value is reported when the BROADCASTMTE input is HIGH. | The reset values can be the following:<br>0b0001, 0b0011,<br>respectively to the value. |
| [7:4]   | SSBS        | Speculative Store Bypassing controls in AArch64 state. Defined values are:<br><br><b>0b0010</b><br>As 0b0001, and adds the MSR and MRS instructions to directly read and write the PSTATE.SSBS field.                                                                                                                                                                                                                                                                                                                                                                                                 | 0b0010                                                                                  |
| [3:0]   | BT          | Branch Target Identification mechanism support in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>The Branch Target Identification mechanism is implemented.                                                                                                                                                                                                                                                                                                                                                                                                                               | 0b0001                                                                                  |

## Access

MRS <Xt>, ID\_AA64PFR1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0100 | 0b001 |

## Accessibility

MRS <Xt>, ID\_AA64PFR1\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64PFR1_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64PFR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64PFR1_EL1;

```

### A.5.6 ID\_AA64ZFR0\_EL1, SVE Feature ID Register 0

Provides additional information about the implemented features of the AArch64 Scalable Vector Extension instruction set, when one or more of FEAT\_SVE and FEAT\_SME is implemented.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations



Note

Prior to the introduction of the features described by this register, this register was unnamed and reserved, RES0 from EL1, EL2, and EL3.

If FEAT\_SME is implemented and FEAT\_SVE is not implemented, then SVE instructions can only be executed when the PE is in Streaming SVE mode and the instructions are legal to execute in Streaming SVE mode.

#### Attributes

##### Width

64

##### Functional group

Identification registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | 0000 | 0000 | xxxx | 0001 | xxxx | xxxx | xxxx | xxxx | 0000 | 0001 | 0001 | xxxx | xxxx | xxxx | 0001 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Note

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-64: AArch64\_id\_aa64zfr0\_el1 bit assignments**

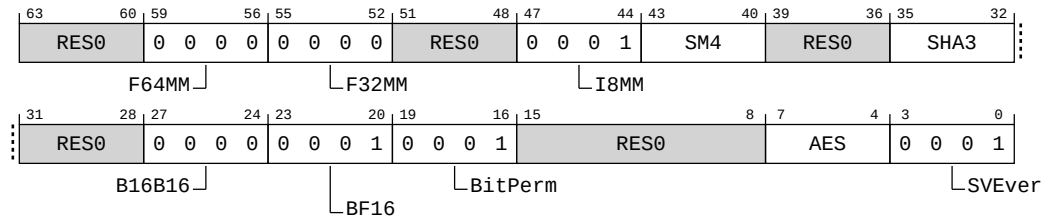


**Table A-191: ID\_AA64ZFR0\_EL1 bit descriptions**

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset                                                                           |
|---------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [63:60] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0                                                                            |
| [59:56] | F64MM | Indicates support for SVE FP64 double-precision floating-point matrix multiplication instructions. Defined values are:<br><br><b>0b0000</b><br>Double-precision matrix multiplication and related SVE instructions are not implemented.                                                                                                                                                                                                                                                                       | 0b0000                                                                          |
| [55:52] | F32MM | Indicates support for the SVE FP32 single-precision floating-point matrix multiplication instruction. Defined values are:<br><br><b>0b0000</b><br>Single-precision matrix multiplication instruction is not implemented.                                                                                                                                                                                                                                                                                      | 0b0000                                                                          |
| [51:48] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0                                                                            |
| [47:44] | I8MM  | Indicates support for SVE Int8 matrix multiplication instructions. Defined values are:<br><br><b>0b0001</b><br>SVE SMMLA, SUDOT, UMMLA, USMMLA, and USDOT instructions are implemented.                                                                                                                                                                                                                                                                                                                       | 0b0001                                                                          |
| [43:40] | SM4   | Indicates support for SVE SM4 instructions. Defined values are:<br><br><b>0b0000</b><br>SVE2 SM4 instructions are not implemented. This value is reported when the Cryptographic Extension is not implemented or is disabled, or SM3/SM4 Cryptographic extensions are not implemented or are disabled.<br><br><b>0b0001</b><br>SVE2 SM4E and SM4EKEY instructions are implemented. This value is reported when the Cryptographic Extension is implemented and SM3/SM4 Cryptographic instructions are enabled. | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [39:36] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0                                                                            |
| [35:32] | SHA3  | Indicates support for the SVE SHA3 instructions. Defined values are:<br><br><b>0b0000</b><br>SVE2 SHA-3 instructions are not implemented. This value is reported when Cryptographic extensions are not implemented or are disabled.<br><br><b>0b0001</b><br>SVE2 RAX1 instruction is implemented. This value is reported when Cryptographic extensions are implemented and enabled.                                                                                                                           | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [31:28] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0                                                                            |

| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset                                                                           |
|---------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [27:24] | B16B16      | Indicates support for SVE2.1 non-widening BFloat16 instructions. Defined values are:<br><br><b>0b0000</b><br>SVE2.1 non-widening BFloat16 instructions are not implemented.                                                                                                                                                                                                                                                                                         | 0b0000                                                                          |
| [23:20] | BF16        | Indicates support for SVE BFloat16 instructions. Defined values are:<br><br><b>0b0001</b><br>SVE BFCVT, BFCVTNT, BFDOT, BFMLALB, BFMLALT, and BFMMLA instructions are implemented.                                                                                                                                                                                                                                                                                  | 0b0001                                                                          |
| [19:16] | BitPerm     | Indicates support for SVE bit permute instructions. Defined values are:<br><br><b>0b0001</b><br>SVE BDEP, BEXT, and BGRP instructions are implemented.                                                                                                                                                                                                                                                                                                              | 0b0001                                                                          |
| [15:8]  | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <b>RES0</b>                                                                     |
| [7:4]   | AES         | Indicates support for SVE AES instructions. Defined values are:<br><br><b>0b0000</b><br>SVE2-AES instructions are not implemented. This value is reported when Cryptographic extensions are not implemented or are disabled.<br><br><b>0b0010</b><br>SVE2 AESE, AESD, AESMC, and AESIMC instructions are implemented plus SVE2 PMULLB and PMULLT instructions with 64-bit source. This value is reported when Cryptographic extensions are implemented and enabled. | The reset values can be the following: 0b0000, 0b0010, respective to the value. |
| [3:0]   | SVEver      | Indicates support for SVE instructions when one or more of FEAT_SME and FEAT_SVE is implemented. Defined values are:<br><br><b>0b0000</b><br>The SVE instructions are implemented.<br><br><b>0b0001</b><br>As 0b0000, and adds the mandatory SVE2 instructions.<br><br>For this product, the selected value is 0b0001.                                                                                                                                              | 0b0001                                                                          |

## Access

MRS <Xt>, ID\_AA64ZFR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0100 | 0b100 |

## Accessibility

MRS <Xt>, ID\_AA64ZFR0\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ZFR0_EL1;

```

```

elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ZFR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64ZFR0_EL1;

```

## A.5.7 ID\_AA64DFR0\_EL1, AArch64 Debug Feature Register 0

Provides top level information about the debug system in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

The external register ext-EDDFR gives information from this register.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

0001 0000 0000 1111 0001 0001 1111 0011 0001 0000 0011 0000 0101 0111 0001  
1001

### Bit descriptions

Figure A-65: AArch64\_id\_aa64dfr0\_el1 bit assignments

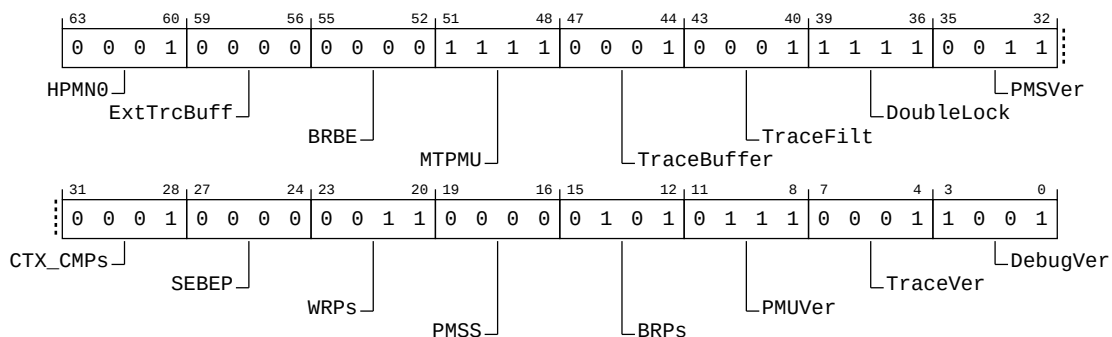


**Table A-193: ID\_AA64DFR0\_EL1 bit descriptions**

| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset  |
|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | HPMN0       | Zero PMU event counters for a Guest operating system. Defined values are:<br><b>0b0001</b><br>Setting AArch64-MDCR_EL2.HPMN to zero has defined behavior.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0b0001 |
| [59:56] | ExtTrcBuff  | Trace Buffer External Mode Extension. Defined values are:<br><b>0b0000</b><br>Trace Buffer External Mode not implemented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0b0000 |
| [55:52] | BRBE        | Branch Record Buffer Extension. Defined values are:<br><b>0b0000</b><br>Branch Record Buffer Extension not implemented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0b0000 |
| [51:48] | MTPMU       | Multi-threaded PMU extension. Defined values are:<br><b>0b1111</b><br>FEAT_MTPMU not implemented. If FEAT_PMUv3 is implemented, AArch64-PMEVTYPER<n>_ELO.MT and AArch32-PMEVTYPER<n>.MT are RES0.                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0b1111 |
| [47:44] | TraceBuffer | Trace Buffer Extension. Defined values are:<br><b>0b0001</b><br>Trace Buffer Extension implemented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0b0001 |
| [43:40] | TraceFilt   | Armv8.4 Self-hosted Trace Extension version. Defined values are:<br><b>0b0001</b><br>Armv8.4 Self-hosted Trace Extension implemented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0b0001 |
| [39:36] | DoubleLock  | OS Double Lock implemented. Defined values are:<br><b>0b1111</b><br>OS Double Lock not implemented. AArch64-OSDLR_EL1 is <b>RAZ/WI</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0b1111 |
| [35:32] | PMSVer      | Statistical Profiling Extension version. Defined values are:<br><b>0b0011</b><br>As 0b0010, and adds: <ul style="list-style-type: none"> <li>Discard mode.</li> <li>Extended event filtering, including the AArch64-PMSNEVFR_EL1 System register.</li> <li>Support for the <b>OPTIONAL</b> previous branch target Address packet.</li> <li>If FEAT_PMUv3 is implemented, controls to freeze the PMU event counters after an SPE buffer management event occurs.</li> <li>If FEAT_PMUv3 is implemented, the SAMPLE_FEED_BR, SAMPLE_FEED_EVENT, SAMPLE_FEED_LAT, SAMPLE_FEED_LD, SAMPLE_FEED_OP, and SAMPLE_FEED_ST PMU events.</li> </ul> | 0b0011 |
| [31:28] | CTX_CMPs    | Number of breakpoints that are context-aware, minus 1.<br><b>0b0001</b><br>Two context-aware breakpoints are included                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0b0001 |
| [27:24] | SEBEP       | Synchronous-exception-based event profiling. Defined values are:<br><b>0b0000</b><br>Synchronous-exception-based event profiling not implemented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0b0000 |
| [23:20] | WRPs        | Number of watchpoints, minus 1.<br><b>0b0011</b><br>Four Watchpoints                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b0011 |

| Bits    | Name     | Description                                                                                                                                                                     | Reset  |
|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [19:16] | PMSS     | PMU Snapshot extension. Defined values are:<br><br><b>0b0000</b><br>PMU snapshot extension not implemented.                                                                     | 0b0000 |
| [15:12] | BRPs     | Number of breakpoints, minus 1.<br><br><b>0b0101</b><br>Six Breakpoints                                                                                                         | 0b0101 |
| [11:8]  | PMUVer   | Performance Monitors Extension version. Defined value is:<br><br><b>0b0111</b><br>Performance Monitors Extension implemented, PMUv3 for Armv8.7                                 | 0b0111 |
| [7:4]   | TraceVer | Trace support. Indicates whether System register interface to a trace unit is implemented. Defined values are:<br><br><b>0b0001</b><br>Trace unit System registers implemented. | 0b0001 |
| [3:0]   | DebugVer | Debug architecture version. Indicates presence of Armv8 debug architecture. Defined values are:<br><br><b>0b1001</b><br>Armv8.4 debug architecture, FEAT_Debugv8p4.             | 0b1001 |

## Access

MRS <Xt>, ID\_AA64DFR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0101 | 0b000 |

## Accessibility

MRS <Xt>, ID\_AA64DFR0\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64DFR0_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64DFR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64DFR0_EL1;

```

## A.5.8 ID\_AA64DFR1\_EL1, AArch64 Debug Feature Register 1

Provides top level information about the debug system in AArch64.

## Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | 0000 | 0000 | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-66: AArch64\_id\_aa64dfr1\_el1 bit assignments

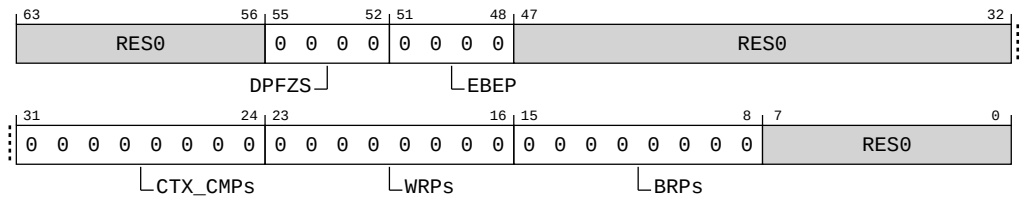


Table A-195: ID\_AA64DFR1\_EL1 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                       | Reset  |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:56] | RES0     | Reserved                                                                                                                                                                                                                          | RES0   |
| [55:52] | DPFZS    | Behavior of the cycle counter when event counting is frozen by a Statistical Profiling management event. Defined values are:<br><b>0b0000</b><br>The cycle counter AArch64-PMCCNTR_EL0 is never affected by AArch64-PMCR_EL0.FZS. | 0b0000 |
| [51:48] | EBEP     | Exception-based event profiling. Defined values are:<br><b>0b0000</b><br>Exception-based event profiling not implemented.                                                                                                         | 0b0000 |
| [47:32] | RES0     | Reserved                                                                                                                                                                                                                          | RES0   |
| [31:24] | CTX_CMPS | Context-aware breakpoints. Defined values are:<br><b>0b00000000</b><br>AArch64-ID_AA64DFR0_EL1.CTX_CMPS is the number of breakpoints that are context-aware, minus 1.                                                             | 0x00   |



| Bits    | Name | Description                                                                                                                      | Reset |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------|-------|
| [23:16] | WRPs | Watchpoints. Defined values are:<br><br><b>0b00000000</b><br>AArch64-ID_AA64DFR0_EL1.WRPs is the number of watchpoints, minus 1. | 0x00  |
| [15:8]  | BRPs | Breakpoints. Defined values are:<br><br><b>0b00000000</b><br>AArch64-ID_AA64DFR0_EL1.BRPs is the number of breakpoints, minus 1. | 0x00  |
| [7:0]   | RES0 | Reserved                                                                                                                         | RES0  |

### Access

MRS <Xt>, ID\_AA64DFR1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0101 | 0b001 |

### Accessibility

MRS <Xt>, ID\_AA64DFR1\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64DFR1_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64DFR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64DFR1_EL1;

```

## A.5.9 ID\_AA64AFR0\_EL1, AArch64 Auxiliary Feature Register 0

Provides information about the **IMPLEMENTATION DEFINED** features of the PE in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

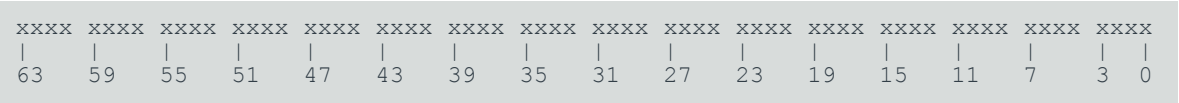
#### Functional group

Identification registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-67: AArch64\_id\_aa64afr0\_el1 bit assignments

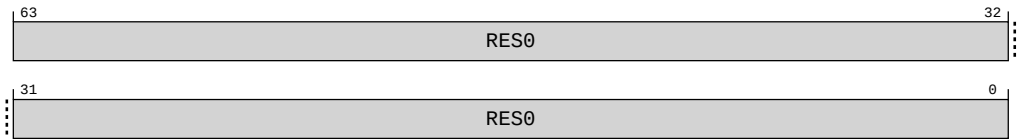


Table A-197: ID\_AA64AFR0\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, ID\_AA64AFR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0101 | 0b100 |

Accessibility

MRS <Xt>, ID\_AA64AFR0\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64AFR0_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64AFR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64AFR0_EL1;
```

A.5.10 ID\_AA64AFR1\_EL1, AArch64 Auxiliary Feature Register 1

Reserved for future expansion of information about the **IMPLEMENTATION DEFINED** features of the PE in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

64

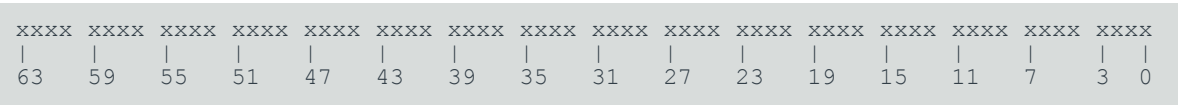
Functional group

Identification registers

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-68: AArch64\_id\_aa64afr1\_el1 bit assignments

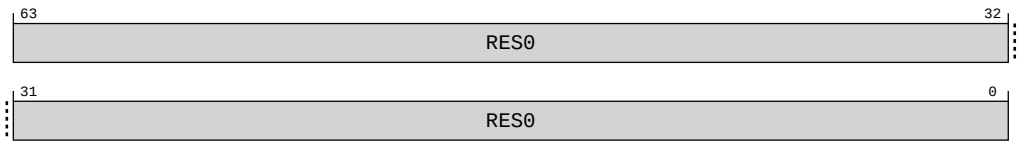


Table A-199: ID\_AA64AFR1\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, ID\_AA64AFR1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0101 | 0b101 |

Accessibility

MRS <Xt>, ID\_AA64AFR1\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64AFR1_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64AFR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64AFR1_EL1;
```

A.5.11 ID\_AA64ISAR0\_EL1, AArch64 Instruction Set Attribute Register 0

Provides information about the instructions implemented in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0000 | 0010 | 0010 | 0001 | 0001 | xxxx | xxxx | xxxx | 0001 | 0000 | 0010 | 0001 | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-69: AArch64\_id\_aa64isar0\_el1 bit assignments**

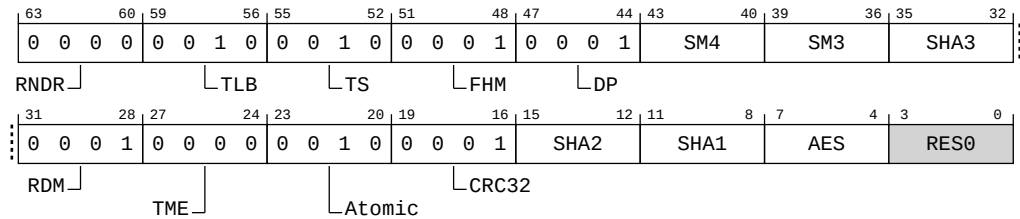


**Table A-201: ID\_AA64ISAR0\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                          | Reset  |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | RNDR | Indicates support for Random Number instructions in AArch64 state.<br><br>When FEAT_RNG_TRAP is implemented, the value returned by a direct read of ID_AA64ISAR0_EL1.RNDR is further controlled by the value of AArch64-SCR_EL3.TRNDR.<br><br>Defined values are:<br><b>0b0000</b><br>No Random Number instructions are implemented. | 0b0000 |
| [59:56] | TLB  | Indicates support for Outer Shareable and TLB range maintenance instructions. Defined values are:<br><b>0b0010</b><br>Outer Shareable and TLB range maintenance instructions are implemented.                                                                                                                                        | 0b0010 |
| [55:52] | TS   | Indicates support for flag manipulation instructions. Defined values are:<br><b>0b0010</b><br>CFINV, RMIF, SETF16, SETF8, AXFLAG, and XAFLAG instructions are implemented.                                                                                                                                                           | 0b0010 |
| [51:48] | FHM  | Indicates support for FMLAL and FMLSL instructions. Defined values are:<br><b>0b0001</b><br>FMLAL and FMLSL instructions are implemented.                                                                                                                                                                                            | 0b0001 |
| [47:44] | DP   | Indicates support for Dot Product instructions in AArch64 state. Defined values are:<br><b>0b0001</b><br>UDOT and SDOT instructions implemented.                                                                                                                                                                                     | 0b0001 |

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset                                                                           |
|---------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [43:40] | SM4    | <p>Indicates support for SM4 instructions in AArch64 state. Defined values are:</p> <p><b>0b0000</b></p> <p>When the Cryptographic Extension is not implemented or is disabled or the SM3/SM4 Cryptographic instructions are disabled, then SM4 instructions are not implemented.</p> <p><b>0b0001</b></p> <p>When the Cryptographic Extension is implemented and the SM3/SM4 Cryptographic instructions are enabled, then SM4 instructions SM4E and SM4EKEY are implemented.</p>                                                     | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [39:36] | SM3    | <p>Indicates support for SM3 instructions in AArch64 state. Defined values are:</p> <p><b>0b0000</b></p> <p>When the Cryptographic Extension is not implemented or is disabled or the SM3/SM4 Cryptographic instructions are disabled, then SM3 instructions are not implemented.</p> <p><b>0b0001</b></p> <p>When the Cryptographic Extension is implemented and the SM3/SM4 Cryptographic instructions are enabled, then SM3 instructions SM3SS1, SM3TT1A, SM3TT1B, SM3TT2A, SM3TT2B, SM3PARTW1, and SM3PARTW2 are implemented.</p> | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [35:32] | SHA3   | <p>Indicates support for SHA3 instructions in AArch64 state. Defined values are:</p> <p><b>0b0000</b></p> <p>When Cryptographic extensions are not implemented or disabled then SHA3 instructions are not implemented.</p> <p><b>0b0001</b></p> <p>When Cryptographic extensions are implemented and enabled then SHA3 instructions EOR3, RAX1, XAR, and BCAX are implemented.</p>                                                                                                                                                    | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [31:28] | RDM    | <p>Indicates support for SQRDMLAH and SQRDMLSH instructions in AArch64 state. Defined values are:</p> <p><b>0b0001</b></p> <p>SQRDMLAH and SQRDMLSH instructions implemented.</p>                                                                                                                                                                                                                                                                                                                                                     | 0b0001                                                                          |
| [27:24] | TME    | <p>Indicates support for TME instructions. Defined values are:</p> <p><b>0b0000</b></p> <p>TME instructions are not implemented.</p> <p><b>When PSTATE.EL IN {EL2, EL1}</b></p> <p>Access to this field is: <b>RAZ/WI</b></p> <p><b>When PSTATE.EL == EL1 &amp;&amp; EL2Enabled()</b></p> <p>Access to this field is: <b>RAZ/WI</b></p> <p><b>Otherwise</b></p> <p>Access to this field is: <b>RO</b></p>                                                                                                                             | 0b0000                                                                          |
| [23:20] | Atomic | <p>Indicates support for Atomic instructions in AArch64 state. Defined values are:</p> <p><b>0b0010</b></p> <p>LDADD, LDCLR, LDEOR, LDSET, LDSMAX, LDSMIN, LDUMAX, LDUMIN, CAS, CASP, and SWP instructions implemented.</p>                                                                                                                                                                                                                                                                                                           | 0b0010                                                                          |

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset                                                                           |
|---------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [19:16] | CRC32 | Indicates support for CRC32 instructions in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>CRC32B, CRC32H, CRC32W, CRC32X, CRC32CB, CRC32CH, CRC32CW, and CRC32CX instructions are implemented.                                                                                                                                                                                                                                                                                                                                                                                                                  | 0b0001                                                                          |
| [15:12] | SHA2  | Indicates support for SHA2 instructions in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>When Cryptographic extensions are not implemented or disabled then SHA2 instructions are not implemented.<br><br><b>0b0010</b><br>When Cryptographic extensions are implemented and enabled then SHA256H, SHA256H2, SHA256SU0, SHA256SU1, SHA512H, SHA512H2, SHA512SU0, and SHA512SU1 instructions are implemented.<br><br>When the CRYPTO configuration parameter is true and the CRYPTODISABLE input is low at reset Cryptographic Extensions are implemented                                                        | The reset values can be the following: 0b0000, 0b0010, respective to the value. |
| [11:8]  | SHA1  | Indicates support for SHA1 instructions in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>When Cryptographic extensions are not implemented or disabled then SHA1 instructions are not implemented.<br><br><b>0b0001</b><br>When Cryptographic extensions are implemented and enabled then SHA1C, SHA1P, SHA1M, SHA1H, SHA1SU0, and SHA1SU1 instructions are implemented.<br><br>When the CRYPTO configuration parameter is true and the CRYPTODISABLE input is low at reset Cryptographic Extensions are implemented                                                                                            | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [7:4]   | AES   | Indicates support for AES instructions in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>SVE2-AES instructions are not implemented. This value is reported when Cryptographic extensions are not implemented or are disabled.<br><br><b>0b0010</b><br>SVE2 AESE, AESD, AESMC, and AESIMC instructions are implemented plus SVE2 PMULLB and PMULLT instructions with 64-bit source. This value is reported when Cryptographic extensions are implemented and enabled.<br><br>When the CRYPTO configuration parameter is true and the CRYPTODISABLE input is low at reset Cryptographic Extensions are implemented | The reset values can be the following: 0b0000, 0b0010, respective to the value. |
| [3:0]   | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RES0                                                                            |

## Access

MRS <Xt>, ID\_AA64ISAR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0110 | 0b000 |

## Accessibility

MRS <Xt>, ID\_AA64ISAR0\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ISAR0_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ISAR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64ISAR0_EL1;

```

### A.5.12 ID\_AA64ISAR1\_EL1, AArch64 Instruction Set Attribute Register 1

Provides information about the features and instructions implemented in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Functional group

Identification registers

### Access type

See bit descriptions

### Reset value

0000 0001 0001 0001 0001 0001 0001 0001 0000 0000 0010 0001 0001 0000 0000  
0010



## Bit descriptions

**Figure A-70: AArch64\_id\_aa64isar1\_el1 bit assignments**

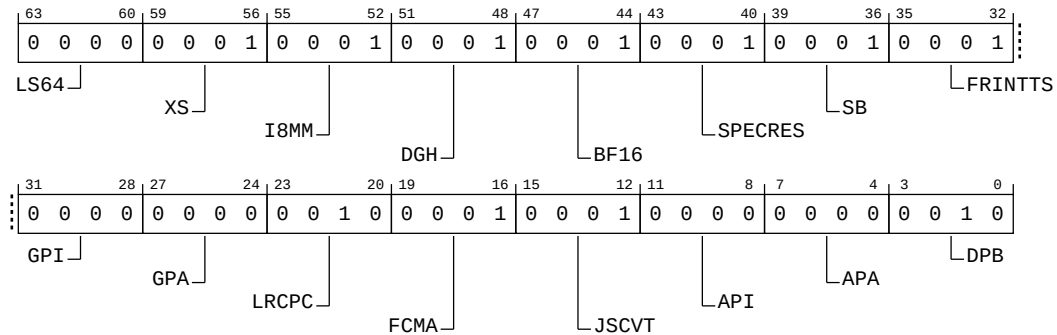


**Table A-203: ID\_AA64ISAR1\_EL1 bit descriptions**

| Bits    | Name    | Description                                                                                                                                                                                                                                                                                                                                      | Reset  |
|---------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | LS64    | Indicates support for LD64B and ST64B* instructions, and the AArch64-ACCDATA_EL1 register. Defined values of this field are:<br><br><b>0b0000</b><br>The LD64B, ST64B, ST64BV, and ST64BV0 instructions, the AArch64-ACCDATA_EL1 register, and associated traps are not supported.                                                               | 0b0000 |
| [59:56] | XS      | Indicates support for the XS attribute, the TLBI and DSB instructions with the nXS qualifier, and the AArch64-HCRX_EL2.{FGTnXS, FnXS} fields in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>The XS attribute, the TLBI and DSB instructions with the nXS qualifier, and the AArch64-HCRX_EL2.{FGTnXS, FnXS} fields are supported. | 0b0001 |
| [55:52] | I8MM    | Indicates support for Advanced SIMD and Floating-point Int8 matrix multiplication instructions in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>SMMLA, SUDOT, UMMLA, USMMLA, and USDOT instructions are implemented.                                                                                                                | 0b0001 |
| [51:48] | DGH     | Indicates support for the Data Gathering Hint instruction. Defined values are:<br><br><b>0b0001</b><br>Data Gathering Hint is implemented.                                                                                                                                                                                                       | 0b0001 |
| [47:44] | BF16    | Indicates support for Advanced SIMD and Floating-point BFloat16 instructions in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>BFCVT, BFCVTN, BFCVTN2, BFDOT, BFMLALB, BFMLALT, and BFMMMLA instructions are implemented.                                                                                                            | 0b0001 |
| [43:40] | SPECRES | Indicates support for prediction invalidation instructions in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>CFP RCTX, DVP RCTX and CPP RCTX instructions are implemented.                                                                                                                                                           | 0b0001 |
| [39:36] | SB      | Indicates support for SB instruction in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>SB instruction is implemented.                                                                                                                                                                                                                | 0b0001 |

| Bits    | Name    | Description                                                                                                                                                                                                                                                                                                                                                           | Reset  |
|---------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [35:32] | FRINTTS | Indicates support for the FRINT32Z, FRINT32X, FRINT64Z, and FRINT64X instructions are implemented. Defined values are:<br><br><b>0b0001</b><br>FRINT32Z, FRINT32X, FRINT64Z, and FRINT64X instructions are implemented.                                                                                                                                               | 0b0001 |
| [31:28] | GPI     | Indicates support for an <b>IMPLEMENTATION DEFINED</b> algorithm is implemented in the PE for generic code authentication in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>Generic Authentication using an <b>IMPLEMENTATION DEFINED</b> algorithm is not implemented.                                                                                   | 0b0000 |
| [27:24] | GPA     | Indicates whether the QARMA5 algorithm is implemented in the PE for generic code authentication in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>Generic Authentication using the QARMA5 algorithm is not implemented.                                                                                                                                   | 0b0000 |
| [23:20] | LRCPC   | Indicates support for weaker release consistency, RCpc, based model. Defined values are:<br><br><b>0b0010</b><br>As 0b0001, and the LDAPR (unscaled immediate) and STLR (unscaled immediate) instructions are implemented.                                                                                                                                            | 0b0010 |
| [19:16] | FCMA    | Indicates support for complex number addition and multiplication, where numbers are stored in vectors. Defined values are:<br><br><b>0b0001</b><br>The FCMLA and FCADD instructions are implemented.                                                                                                                                                                  | 0b0001 |
| [15:12] | JSCVT   | Indicates support for JavaScript conversion from double precision floating point values to integers in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>The FJCVTZS instruction is implemented.                                                                                                                                                             | 0b0001 |
| [11:8]  | API     | Indicates whether an <b>IMPLEMENTATION DEFINED</b> algorithm is implemented in the PE for address authentication, in AArch64 state. This applies to all Pointer Authentication instructions other than the PACGA instruction. Defined values are:<br><br><b>0b0000</b><br>Address Authentication using an <b>IMPLEMENTATION DEFINED</b> algorithm is not implemented. | 0b0000 |
| [7:4]   | APA     | Indicates whether the QARMA5 algorithm is implemented in the PE for address authentication, in AArch64 state. This applies to all Pointer Authentication instructions other than the PACGA instruction. Defined values are:<br><br><b>0b0000</b><br>Address Authentication using the QARMA5 algorithm is not implemented.                                             | 0b0000 |
| [3:0]   | DPB     | Data Persistence writeback. Indicates support for the DC CVAP and DC CVADP instructions in AArch64 state. Defined values are:<br><br><b>0b0010</b><br>DC CVAP and DC CVADP supported.                                                                                                                                                                                 | 0b0010 |

## Access

MRS <Xt>, ID\_AA64ISAR1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0110 | 0b001 |

Accessibility

MRS <Xt>, ID\_AA64ISAR1\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ISAR1_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ISAR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64ISAR1_EL1;
```

A.5.13 ID\_AA64ISAR2\_EL1, AArch64 Instruction Set Attribute Register 2

Provides information about the features and instructions implemented in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations



Note

Prior to the introduction of the features described by this register, this register was unnamed and reserved, RES0 from EL1, EL2, and EL3.

Attributes

Width

64

Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | 0000 | 0000 | xxxx | 0000 | 0000 | 0000 | 0000 | 0001 | 0000 | 0000 | 0101 | 0001 | 0000 | 0010 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-71: AArch64\_id\_aa64isar2\_el1 bit assignments**

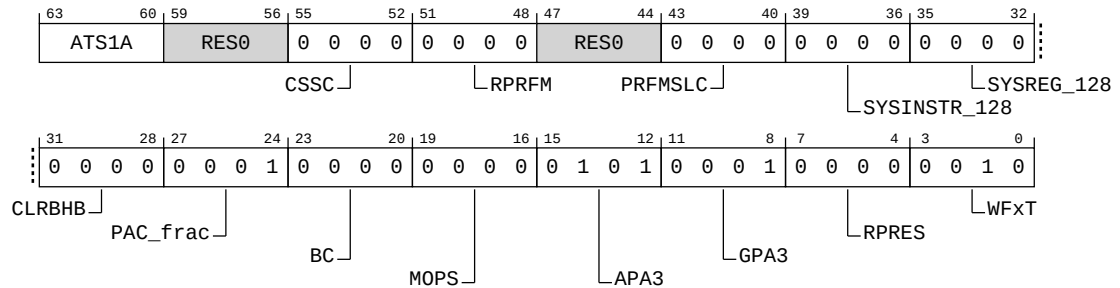


**Table A-205: ID\_AA64ISAR2\_EL1 bit descriptions**

| Bits    | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                    | Reset                                                                           |
|---------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [63:60] | ATS1A   | Indicates support for address translation instructions, which perform stage 1 address translation for the given virtual address without checking for stage 1 permissions. Defined values are:<br><b>0b0000</b><br>Address Translate Stage 1 instructions without Permissions Checks are not implemented<br><b>0b0001</b><br>Address Translate Stage 1 instructions without Permissions Checks are implemented. | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [59:56] | RES0    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                       | RES0                                                                            |
| [55:52] | CSSC    | Indicates support for common short sequence compression instructions. Defined values are:<br><b>0b0000</b><br>Common short sequence compression instructions are not implemented.                                                                                                                                                                                                                              | 0b0000                                                                          |
| [51:48] | RPRFM   | RPRFM hint instruction. Defined values are:<br><b>0b0000</b><br>RPRFM hint instruction is not implemented and is treated as a <b>NOP</b> .                                                                                                                                                                                                                                                                     | 0b0000                                                                          |
| [47:44] | RES0    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                       | RES0                                                                            |
| [43:40] | PRFMslc | Indicates whether the PRFM instructions support a system level cache option. Defined values are:<br><b>0b0000</b><br>The PRFM instructions do not support the SLC target.                                                                                                                                                                                                                                      | 0b0000                                                                          |

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset  |
|---------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [39:36] | SYSINSTR_128 | SYSINSTR_128. Indicates support for System instructions that can take 128-bit inputs. Defined values are:<br><br><b>0b0000</b><br>System instructions that can take 128-bit inputs are not supported.                                                                                                                                                                                                                                                                                                                        | 0b0000 |
| [35:32] | SYSREG_128   | SYSREG_128. Indicates support for instructions to access 128-bit System Registers. Defined values are:<br><br><b>0b0000</b><br>Instructions to access 128-bit System Registers are not supported.                                                                                                                                                                                                                                                                                                                            | 0b0000 |
| [31:28] | CLRBHB       | Indicates support for the CLRBHB instruction in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>CLRBHB instruction is not implemented.                                                                                                                                                                                                                                                                                                                                                                            | 0b0000 |
| [27:24] | PAC_frac     | Indicates whether the ConstPACField() function used as part of the PAC addition returns FALSE or TRUE.<br><br><b>0b0001</b><br>ConstPACField() returns TRUE.                                                                                                                                                                                                                                                                                                                                                                 | 0b0001 |
| [23:20] | BC           | Indicates support for the BC instruction in AArch64 state. Defined values are:<br><br><b>0b0000</b><br>BC instruction is not implemented.                                                                                                                                                                                                                                                                                                                                                                                    | 0b0000 |
| [19:16] | MOPS         | Indicates support for the Memory Copy and Memory Set instructions in AArch64 state.<br><br><b>0b0000</b><br>The Memory Copy and Memory Set instructions are not implemented in AArch64 state.                                                                                                                                                                                                                                                                                                                                | 0b0000 |
| [15:12] | APA3         | Indicates whether the QARMA3 algorithm is implemented in the PE for address authentication in AArch64 state. This applies to all Pointer Authentication instructions other than the <b>PACGA</b> instruction. Defined values are:<br><br><b>0b0101</b><br>Address Authentication using the QARMA3 algorithm is implemented, with the HaveEnhancedPAC2() function returning TRUE, the HaveFPAC() function returning TRUE, the HaveFPACCombined() function returning TRUE, and the HaveEnhancedPAC() function returning FALSE. | 0b0101 |
| [11:8]  | GPA3         | Indicates whether the QARMA3 algorithm is implemented in the PE for generic code authentication in AArch64 state. Defined values are:<br><br><b>0b0001</b><br>Generic Authentication using the QARMA3 algorithm is implemented. This includes the <b>PACGA</b> instruction.                                                                                                                                                                                                                                                  | 0b0001 |
| [7:4]   | RPRES        | Indicates support for 12 bits of mantissa in reciprocal and reciprocal square root instructions in AArch64 state, when AArch64-FPCR.AH is 1. Defined values are:<br><br><b>0b0000</b><br>Reciprocal and reciprocal square root estimates give 8 bits of mantissa, when AArch64-FPCR.AH is 1.                                                                                                                                                                                                                                 | 0b0000 |
| [3:0]   | WFXT         | Indicates support for the <b>WFET</b> and <b>WFIT</b> instructions in AArch64 state. Defined values are:<br><br><b>0b0010</b><br><b>WFET</b> and <b>WFIT</b> are supported, and the register number is reported in the ESR_ELx on exceptions.                                                                                                                                                                                                                                                                                | 0b0010 |

Access

MRS <Xt>, ID\_AA64ISAR2\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0110 | 0b010 |

Accessibility

MRS <Xt>, ID\_AA64ISAR2\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ISAR2_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ISAR2_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64ISAR2_EL1;
```

A.5.14 ID\_AA64MMFR0\_EL1, AArch64 Memory Model Feature Register 0

Provides information about the implemented memory model and memory management support in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0010 | 0001 | xxxx | xxxx | 0000 | 0010 | 0010 | 0010 | 0000 | 0000 | 0001 | xxxx | 0001 | 0001 | 0010 | 0010 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-72: AArch64\_id\_aa64mmfr0\_el1 bit assignments**

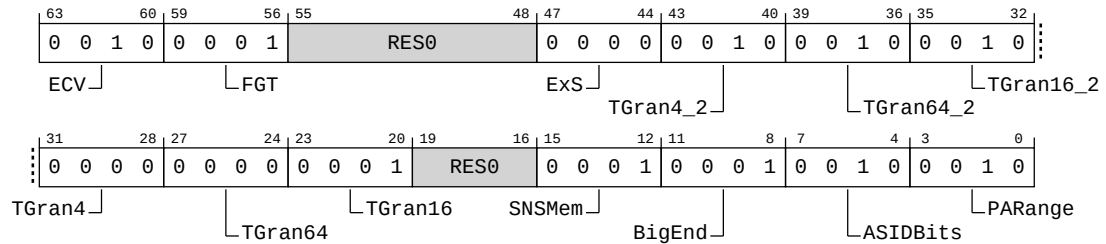


**Table A-207: ID\_AA64MMFR0\_EL1 bit descriptions**

| Bits    | Name      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset  |
|---------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | ECV       | Indicates presence of Enhanced Counter Virtualization. Defined values are:<br><b>0b0010</b><br>As 0b0001, and also includes support for AArch64-CNTHCTL_EL2.ECV and AArch64-CNTPOFF_EL2.                                                                                                                                                                                                                                                                                                                                                                                                | 0b0010 |
| [59:56] | FGT       | Indicates presence of the Fine-Grained Trap controls. Defined values are:<br><b>0b0001</b><br>Fine-grained trap controls are implemented. Supports: <ul style="list-style-type: none"> <li>If EL2 is implemented, the AArch64-HAFGRTR_EL2, AArch64-HDFGRTR_EL2, AArch64-HDFGWTR_EL2, AArch64-HFGRTR_EL2, AArch64-HFGITR_EL2 and AArch64-HFGWTR_EL2 registers, and their associated traps.</li> <li>If EL2 is implemented, AArch64-MDCR_EL2.TDCC.</li> <li>If EL3 is implemented, AArch64-MDCR_EL3.TDCC.</li> <li>If both EL2 and EL3 are implemented, AArch64-SCR_EL3.FGTEn.</li> </ul> | 0b0001 |
| [55:48] | RES0      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RES0   |
| [47:44] | ExS       | Indicates support for disabling context synchronizing exception entry and exit. Defined values are:<br><b>0b0000</b><br>All exception entries and exits are context synchronization events.                                                                                                                                                                                                                                                                                                                                                                                             | 0b0000 |
| [43:40] | TGran4_2  | Indicates support for 4KB memory granule size at stage 2. Defined values are:<br><b>0b0010</b><br>4KB granule supported at stage 2.                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b0010 |
| [39:36] | TGran64_2 | Indicates support for 64KB memory granule size at stage 2. Defined values are:<br><b>0b0010</b><br>64KB granule supported at stage 2.                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0b0010 |

| Bits    | Name        | Description                                                                                                                                                                            | Reset       |
|---------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [35:32] | TGran16_2   | Indicates support for 16KB memory granule size at stage 2. Defined values are:<br><br><b>0b0010</b><br>16KB granule supported at stage 2.                                              | 0b0010      |
| [31:28] | TGran4      | Indicates support for 4KB memory translation granule size. Defined values are:<br><br><b>0b0000</b><br>4KB granule supported.                                                          | 0b0000      |
| [27:24] | TGran64     | Indicates support for 64KB memory translation granule size. Defined values are:<br><br><b>0b0000</b><br>64KB granule supported.                                                        | 0b0000      |
| [23:20] | TGran16     | Indicates support for 16KB memory translation granule size. Defined values are:<br><br><b>0b0001</b><br>16KB granule supported.                                                        | 0b0001      |
| [19:16] | <b>RES0</b> | Reserved                                                                                                                                                                               | <b>RES0</b> |
| [15:12] | SNSMem      | Indicates support for a distinction between Secure and Non-secure Memory. Defined values are:<br><br><b>0b0001</b><br>Does support a distinction between Secure and Non-secure Memory. | 0b0001      |
| [11:8]  | BigEnd      | Indicates support for mixed-endian configuration. Defined values are:<br><br><b>0b0001</b><br>Mixed-endian support. The SCTLR_ELx.EE and SCTLR_EL1.EOE bits can be configured.         | 0b0001      |
| [7:4]   | ASIDBits    | Number of ASID bits. Defined values are:<br><br><b>0b0010</b><br>16 bits.                                                                                                              | 0b0010      |
| [3:0]   | PARange     | Physical Address range supported. Defined values are:<br><br><b>0b0010</b><br>40 bits, 1TB.                                                                                            | 0b0010      |

## Access

MRS <Xt>, ID\_AA64MMFR0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0111 | 0b000 |

## Accessibility

MRS <Xt>, ID\_AA64MMFR0\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64MMFR0_EL1;
elseif PSTATE.EL == EL2 then

```



```

X[t, 64] = ID_AA64MMFR0_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64MMFR0_EL1;

```

## A.5.15 ID\_AA64MMFR1\_EL1, AArch64 Memory Model Feature Register 1

Provides information about the implemented memory model and memory management support in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

0001 0000 0000 0001 0001 0001 0001 0000 0001 0000 0011 0001 0010 0001 0010 0010

### Bit descriptions

**Figure A-73: AArch64\_id\_aa64mmfr1\_el1 bit assignments**

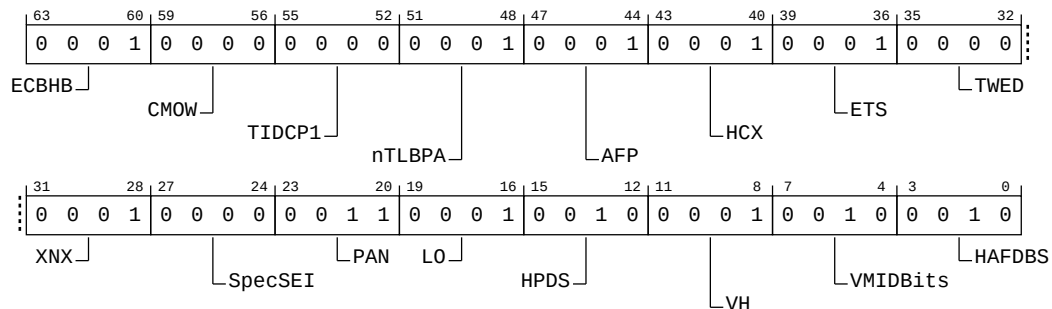


**Table A-209: ID\_AA64MMFR1\_EL1 bit descriptions**

| Bits    | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Reset  |
|---------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | ECBHB   | Indicates support for restrictions on branch history speculation around exceptions. Defined values are:<br><b>0b0001</b><br>The branch history information created in a context before an exception to a higher Exception level using AArch64 cannot be used by code before that exception to exploitatively control the execution of any indirect branches in code in a different context after the exception. | 0b0001 |
| [59:56] | CMOW    | Indicates support for cache maintenance instruction permission. Defined values are:<br><b>0b0000</b><br>AArch64-SCTLR_EL1.CMOW, AArch64-SCTLR_EL2.CMOW, and AArch64-HCRX_EL2.CMOW bits are not implemented.                                                                                                                                                                                                     | 0b0000 |
| [55:52] | TIDCP1  | Indicates whether AArch64-SCTLR_EL1.TIDCP and AArch64-SCTLR_EL2.TIDCP are implemented in AArch64 state. Defined values are:<br><b>0b0000</b><br>AArch64-SCTLR_EL1.TIDCP and AArch64-SCTLR_EL2.TIDCP bits are not implemented and are <b>RES0</b> .                                                                                                                                                              | 0b0000 |
| [51:48] | nTLBPA  | Indicates support for intermediate caching of translation table walks. Defined values are:<br><b>0b0001</b><br>The intermediate caching of translation table walks does not include non-coherent physical translation caches.                                                                                                                                                                                   | 0b0001 |
| [47:44] | AFP     | Indicates support for AArch64-FPCR.{AH, FIZ, NEP}. Defined values are:<br><b>0b0001</b><br>The AArch64-FPCR.{AH, FIZ, NEP} fields are supported.                                                                                                                                                                                                                                                                | 0b0001 |
| [43:40] | HCX     | Indicates support for AArch64-HCRX_EL2 and its associated EL3 trap. Defined values are:<br><b>0b0001</b><br>AArch64-HCRX_EL2 and its associated EL3 trap are supported.                                                                                                                                                                                                                                         | 0b0001 |
| [39:36] | ETS     | Indicates support for Enhanced Translation Synchronization. Defined values are:<br><b>0b0001</b><br>Enhanced Translation Synchronization is not supported.                                                                                                                                                                                                                                                      | 0b0001 |
| [35:32] | TWED    | Indicates support for the configurable delayed trapping of WFE. Defined values are:<br><b>0b0000</b><br>Configurable delayed trapping of WFE is not supported.                                                                                                                                                                                                                                                  | 0b0000 |
| [31:28] | XNX     | Indicates support for execute-never control distinction by Exception level at stage 2. Defined values are:<br><b>0b0001</b><br>Distinction between ELO and EL1 execute-never control at stage 2 supported.                                                                                                                                                                                                      | 0b0001 |
| [27:24] | SpecSEI | Describes whether the PE can generate SError interrupt exceptions from speculative reads of memory, including speculative instruction fetches.<br><b>0b0000</b><br>The PE never generates an SError interrupt due to an External abort on a speculative read.                                                                                                                                                   | 0b0000 |
| [23:20] | PAN     | Privileged Access Never. Indicates support for the PAN bit in PSTATE, AArch64-SPSR_EL1, AArch64-SPSR_EL2, AArch64-SPSR_EL3, and AArch64-DSPSR_ELO. Defined values are:<br><b>0b0011</b><br>PAN supported, AT S1E1RP and AT S1E1WP instructions supported, and AArch64-SCTLR_EL1.EPAN and AArch64-SCTLR_EL2.EPAN bits supported.                                                                                 | 0b0011 |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset  |
|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [19:16] | LO       | LORegions. Indicates support for LORegions. Defined values are:<br><br><b>0b0001</b><br>LORegions supported.                                                                                                                                                                                                                                                                                                                                                  | 0b0001 |
| [15:12] | HPDS     | Hierarchical Permission Disables. Indicates support for disabling hierarchical controls in translation tables. Defined values are:<br><br><b>0b0010</b><br>Disabling of hierarchical controls supported with the TCR_EL1.{HPD1, HPD0}, TCR_EL2.HPD or TCR_EL2.{HPD1, HPD0}, and TCR_EL3.HPD bits and adds possible hardware allocation of bits[62:59] of the translation table descriptors from the final lookup level for <b>IMPLEMENTATION DEFINED</b> use. | 0b0010 |
| [11:8]  | VH       | Virtualization Host Extensions. Defined values are:<br><br><b>0b0001</b><br>Virtualization Host Extensions supported.                                                                                                                                                                                                                                                                                                                                         | 0b0001 |
| [7:4]   | VMIDBits | Number of VMID bits. Defined values are:<br><br><b>0b0010</b><br>16 bits                                                                                                                                                                                                                                                                                                                                                                                      | 0b0010 |
| [3:0]   | HAFDBS   | Hardware updates to Access flag and Dirty state in translation tables. Defined values are:<br><br><b>0b0010</b><br>As 0b0001, and adds support for hardware update of the Access flag for Block and Page descriptors. Hardware update of dirty state is supported.                                                                                                                                                                                            | 0b0010 |

## Access

MRS <Xt>, ID\_AA64MMFR1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0111 | 0b001 |

## Accessibility

MRS <Xt>, ID\_AA64MMFR1\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64MMFR1_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64MMFR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64MMFR1_EL1;

```

## A.5.16 ID\_AA64MMFR2\_EL1, AArch64 Memory Model Feature Register 2

Provides information about the implemented memory model and memory management support in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations



Note

Prior to the introduction of the features described by this register, this register was unnamed and reserved, RES0 from EL1, EL2, and EL3.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0001 | 0010 | 0010 | 0001 | xxxx | 0001 | 0001 | 0001 | 0001 | 0000 | 0001 | 0000 | 0001 | 0000 | 0001 | 0001 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |



Note

Where the reset reads xxxx, see individual bits.

### Bit descriptions

Figure A-74: AArch64\_id\_aa64mmfr2\_el1 bit assignments

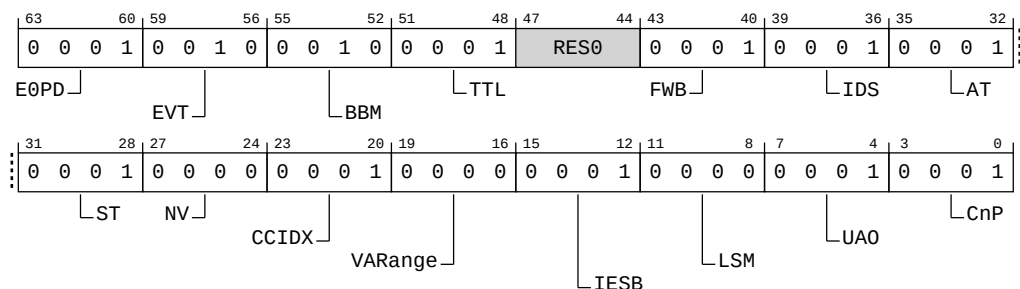


**Table A-211: ID\_AA64MMFR2\_EL1 bit descriptions**

| Bits    | Name    | Description                                                                                                                                                                                                                                                                | Reset  |
|---------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | EOPD    | Indicates support for the EOPD mechanism. Defined values are:<br><br><b>0b0001</b><br>EOPDx mechanism is implemented.                                                                                                                                                      | 0b0001 |
| [59:56] | EVT     | Enhanced Virtualization Traps. If EL2 is implemented, indicates support for the AArch64-HCR_EL2.{TTLBOS, TTLBIS, TOCU, TICAB, TID4} traps. Defined values are:<br><br><b>0b0010</b><br>AArch64-HCR_EL2.{TTLBOS, TTLBIS, TOCU, TICAB, TID4} traps are supported.            | 0b0010 |
| [55:52] | BBM     | Allows identification of the requirements of the hardware to have break-before-make sequences when changing block size for a translation.<br><br><b>0b0010</b><br>Level 2 support for changing block size is supported.                                                    | 0b0010 |
| [51:48] | TTL     | Indicates support for TTL field in address operations. Defined values are:<br><br><b>0b0001</b><br>TLB maintenance instructions by address have bits[47:44] holding the TTL field.                                                                                         | 0b0001 |
| [47:44] | RES0    | Reserved                                                                                                                                                                                                                                                                   | RES0   |
| [43:40] | FWB     | Indicates support for AArch64-HCR_EL2.FWB. Defined values are:<br><br><b>0b0001</b><br>AArch64-HCR_EL2.FWB is supported.                                                                                                                                                   | 0b0001 |
| [39:36] | IDS     | Indicates the value of ESR_ELx.EC that reports an exception generated by a read access to the feature ID space. Defined values are:<br><br><b>0b0001</b><br>All exceptions generated by an AArch64 read access to the feature ID space are reported by ESR_ELx.EC == 0x18. | 0b0001 |
| [35:32] | AT      | Identifies support for unaligned single-copy atomicity and atomic functions. Defined values are:<br><br><b>0b0001</b><br>Unaligned single-copy atomicity and atomic functions with a 16-byte address range aligned to 16-bytes are supported.                              | 0b0001 |
| [31:28] | ST      | Identifies support for small translation tables. Defined values are:<br><br><b>0b0001</b><br>The maximum value of the TCR_ELx.{T0SZ,T1SZ} and VTCR_EL2.T0SZ fields is 48 for 4KB and 16KB granules, and 47 for 64KB granules.                                              | 0b0001 |
| [27:24] | NV      | Nested Virtualization. If EL2 is implemented, indicates support for the use of nested virtualization. Defined values are:<br><br><b>0b0000</b><br>Nested virtualization is not supported.                                                                                  | 0b0000 |
| [23:20] | CCIDX   | Support for the use of revised AArch64-CCSIDR_EL1 register format. Defined values are:<br><br><b>0b0001</b><br>64-bit format implemented for all levels of the CCSIDR_EL1.                                                                                                 | 0b0001 |
| [19:16] | VARange | Indicates support for a larger virtual address. Defined values are:<br><br><b>0b0000</b><br>VMSAv8-64 supports 48-bit VAs.                                                                                                                                                 | 0b0000 |

| Bits    | Name | Description                                                                                                                                                                | Reset  |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:12] | IESB | Indicates support for the IESB bit in the SCTLR_ELx registers. Defined values are:<br><br><b>0b0001</b><br>IESB bit in the SCTLR_ELx registers is supported.               | 0b0001 |
| [11:8]  | LSM  | Indicates support for LSMAOE and nTLSMD bits in AArch64-SCTLR_EL1 and AArch64-SCTLR_EL2. Defined values are:<br><br><b>0b0000</b><br>LSMAOE and nTLSMD bits not supported. | 0b0000 |
| [7:4]   | UAO  | User Access Override. Defined values are:<br><br><b>0b0001</b><br>UAO supported.                                                                                           | 0b0001 |
| [3:0]   | CnP  | Indicates support for Common not Private translations. Defined values are:<br><br><b>0b0001</b><br>Common not Private translations supported.                              | 0b0001 |

## Access

MRS <Xt>, ID\_AA64MMFR2\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0111 | 0b010 |

## Accessibility

MRS <Xt>, ID\_AA64MMFR2\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64MMFR2_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64MMFR2_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64MMFR2_EL1;

```

## A.5.17 ID\_AA64MMFR3\_EL1, AArch64 Memory Model Feature Register 3

Provides information about the implemented memory model and memory management support in AArch64 state.

### Configurations



Prior to the introduction of the features described by this register, this register was unnamed and reserved, RES0 from EL1, EL2, and EL3.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

### Bit descriptions

Figure A-75: AArch64\_id\_aa64mmfr3\_el1 bit assignments

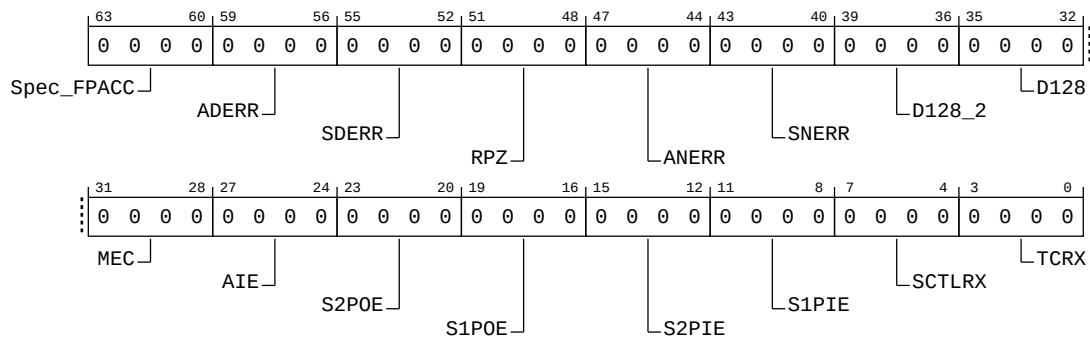


**Table A-213: ID\_AA64MMFR3\_EL1 bit descriptions**

| Bits    | Name       | Description                                                                                                                                                                                                                                                                                                                                                                                                                | Reset  |
|---------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:60] | Spec_FPACC | Speculative behavior in the event of a PAC authentication failure in an implementation that includes FEAT_FPACCOMBINE. Defined values are:<br><br><b>0b0000</b><br>The implementation does not disclose whether the speculative use of pointers processed by a PAC Authentication is materially different in terms of the impact on cached microarchitectural state between passing and failing of the PAC Authentication. | 0b0000 |
| [59:56] | ADERR      | Asynchronous Device error exceptions. With ID_AA64MMFR3_EL1.SDERR, describes the PE behavior for error exceptions on Device memory loads. Defined values are:<br><br><b>0b0000</b><br>If FEAT_RASv2 is not implemented and ID_AA64MMFR3_EL1.SDERR is 0b0000, then the behavior is not described. Otherwise, the behavior is described by ID_AA64MMFR3_EL1.SDERR.                                                           | 0b0000 |
| [55:52] | SDERR      | Synchronous Device error exceptions. With ID_AA64MMFR3_EL1.ADERR, describes the PE behavior for error exceptions on Device memory loads. Defined values are:<br><br><b>0b0000</b><br>If FEAT_RASv2 is not implemented and ID_AA64MMFR3_EL1.ADERR is 0b0000, then the behavior is not described. Otherwise, the behavior is described by ID_AA64MMFR3_EL1.ADERR.                                                            | 0b0000 |
| [51:48] | RPZ        | Remove Poison and Zero. Describes support for the Remove Poison and Zero instruction, DC RPGDZPA. Defined values are:<br><br><b>0b0000</b><br>FEAT_RPZ is not implemented.                                                                                                                                                                                                                                                 | 0b0000 |
| [47:44] | ANERR      | Asynchronous Normal error exceptions. With ID_AA64MMFR3_EL1.SNERR, describes the PE behavior for error exceptions on Normal memory loads. Defined values are:<br><br><b>0b0000</b><br>If FEAT_RASv2 is not implemented and ID_AA64MMFR3_EL1.SNERR is 0b0000, then the behavior is not described. Otherwise, the behavior is described by ID_AA64MMFR3_EL1.SNERR.                                                           | 0b0000 |
| [43:40] | SNERR      | Synchronous Normal error exceptions. With ID_AA64MMFR3_EL1.ANERR, describes the PE behavior for error exceptions on Normal memory loads. Defined values are:<br><br><b>0b0000</b><br>If FEAT_RASv2 is not implemented and ID_AA64MMFR3_EL1.ANERR is 0b0000, then the behavior is not described. Otherwise, the behavior is described by ID_AA64MMFR3_EL1.ANERR.                                                            | 0b0000 |
| [39:36] | D128_2     | 128-bit translation table descriptor at stage 2. Indicates support for 128-bit translation table descriptor at stage 2. Defined values are:<br><br><b>0b0000</b><br>128-bit translation table descriptor Extension at stage 2 is not supported.                                                                                                                                                                            | 0b0000 |
| [35:32] | D128       | 128-bit translation table descriptor. Indicates support for 128-bit translation table descriptor. Defined values are:<br><br><b>0b0000</b><br>128-bit translation table descriptor Extension is not supported.                                                                                                                                                                                                             | 0b0000 |
| [31:28] | MEC        | Indicates support for Memory Encryption Contexts. Defined values are:<br><br><b>0b0000</b><br>Memory Encryption Contexts is not supported.                                                                                                                                                                                                                                                                                 | 0b0000 |
| [27:24] | AIE        | Attribute Indexing. Indicates support for the Attribute Index Enhancement. Defined values are:<br><br><b>0b0000</b><br>The Attribute Index Enhancement is not supported.                                                                                                                                                                                                                                                   | 0b0000 |



| Bits    | Name   | Description                                                                                                                                                                                                        | Reset  |
|---------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [23:20] | S2POE  | Stage 2 Permission Overlay. Indicates support for Permission Overlay at stage 2. Defined values are:<br><br><b>0b0000</b><br>Permission Overlay at stage 2 is not supported.                                       | 0b0000 |
| [19:16] | S1POE  | Stage 1 Permission Overlay. Indicates support for Permission Overlay at stage 1. Defined values are:<br><br><b>0b0000</b><br>Permission Overlay at stage 1 is not supported.                                       | 0b0000 |
| [15:12] | S2PIE  | Stage 2 Permission Indirection. Indicates support for Permission Indirection at stage 2. Defined values are:<br><br><b>0b0000</b><br>Permission Indirection at stage 2 is not supported.                           | 0b0000 |
| [11:8]  | S1PIE  | Stage 1 Permission Indirection. Indicates support for Permission Indirection at stage 1. Defined values are:<br><br><b>0b0000</b><br>Permission Indirection at stage 1 is not supported.                           | 0b0000 |
| [7:4]   | SCTLRX | SCTLRX Extension. Indicates support for Extension of AArch64-SCTLR_EL1. Defined values are:<br><br><b>0b0000</b><br>AArch64-SCTLR2_EL1, AArch64-SCTLR2_EL2 and their associated trap controls are not implemented. | 0b0000 |
| [3:0]   | TCRX   | TCR Extension. Indicates support for Extension of AArch64-TCR_EL1. Defined values are:<br><br><b>0b0000</b><br>AArch64-TCR2_EL1, AArch64-TCR2_EL2 and their associated trap controls are not implemented.          | 0b0000 |

## Access

MRS <Xt>, ID\_AA64MMFR3\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0000 | 0b0111 | 0b011 |

## Accessibility

MRS <Xt>, ID\_AA64MMFR3\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64MMFR3_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64MMFR3_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64MMFR3_EL1;

```

A.5.18 MPAMIDR\_EL1, MPAM ID Register (EL1)

Indicates the presence and maximum PARTID and PMG values supported in the implementation. It also indicates whether the implementation supports MPAM virtualization.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xx10 | 010x | xxxx | xxxx | xxxx | xxxx | 0000 | 0001 | xxxx | xxxx | xxx0 | 011x | 0000 | 0000 | 0011 | 1111 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

MPAMIDR\_EL1 indicates the MPAM implementation parameters of the PE.

Figure A-76: AArch64\_mpamidr\_el1 bit assignments

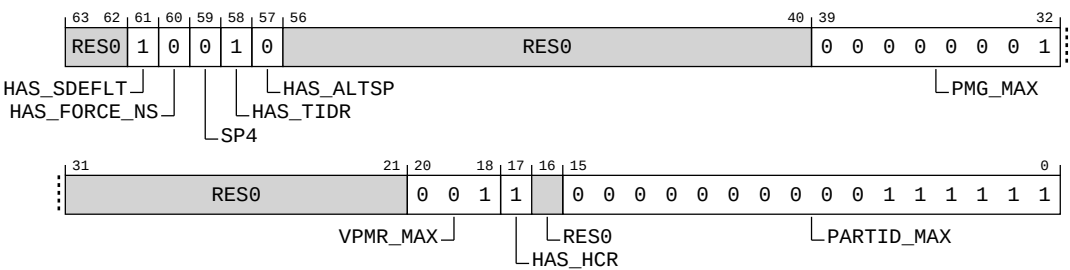


Table A-215: MPAMIDR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:62] | RES0 | Reserved    | RES0  |

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                            | Reset  |
|---------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [61]    | HAS_SDEFLT   | HAS_SDEFLT indicates support for AArch64-MPAM3_EL3.SDEFLT bit. Defined values are:<br><b>0b1</b><br>The SDEFLT bit is implemented in AArch64-MPAM3_EL3.                                                                                                                                                | 0b1    |
| [60]    | HAS_FORCE_NS | HAS_FORCE_NS indicates support for AArch64-MPAM3_EL3.FORCE_NS bit. Defined values are:<br><b>0b0</b><br>The FORCE_NS bit is not implemented in AArch64-MPAM3_EL3.                                                                                                                                      | 0b0    |
| [59]    | SP4          | Supports 4 MPAM PARTID spaces.<br><b>0b0</b><br>MPAM supports 2 PARTID spaces.                                                                                                                                                                                                                         | 0b0    |
| [58]    | HAS_TIDR     | HAS_TIDR indicates support for AArch64-MPAM2_EL2.TIDR bit. Defined values are:<br><b>0b1</b><br>The TIDR bit is implemented in AArch64-MPAM2_EL2.                                                                                                                                                      | 0b1    |
| [57]    | HAS_ALTSP    | HAS_ALTSP indicates support for alternative PARTID spaces.<br><b>0b0</b><br>Alternative PARTID spaces are not implemented.                                                                                                                                                                             | 0b0    |
| [56:40] | RES0         | Reserved                                                                                                                                                                                                                                                                                               | RES0   |
| [39:32] | PMG_MAX      | The largest value of PMG that the implementation can generate. The PMG_I and PMG_D fields of every MPAMn_ELx must implement at least enough bits to represent PMG_MAX.<br><b>0b00000001</b><br>Max PMG field is 1 (1-bit)                                                                              | 0x01   |
| [31:21] | RES0         | Reserved                                                                                                                                                                                                                                                                                               | RES0   |
| [20:18] | VPMR_MAX     | Indicates the maximum register index n for the MPAMVPM<n>_EL2 registers.<br><b>0b001</b><br>2 MPAMVPMn_EL2 registers are implemented                                                                                                                                                                   | 0b001  |
| [17]    | HAS_HCR      | HAS_HCR indicates that the PE implementation supports MPAM virtualization, including AArch64-MPAMHCR_EL2, AArch64-MPAMVPMV_EL2, and MPAMVPM<n>_EL2 with n in the range 0 to VPMR_MAX. Must be 0 if EL2 is not implemented in either Security state.<br><b>0b1</b><br>MPAM virtualization is supported. | 0b1    |
| [16]    | RES0         | Reserved                                                                                                                                                                                                                                                                                               | RES0   |
| [15:0]  | PARTID_MAX   | The largest value of PARTID that the implementation can generate. The PARTID_I and PARTID_D fields of every MPAMn_ELx must implement at least enough bits to represent PARTID_MAX.<br><b>0b0000000000111111</b><br>Max PARTID field is 63                                                              | 0x003F |

## Access

MRS <Xt>, MPAMIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1010 | 0b0100 | 0b100 |

## Accessibility

MRS <Xt>, MPAMIDR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elseif EL2Enabled() && MPAMHCR_EL2.TRAP_MPAMIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MPAM2_EL2.TIDR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = MPAMIDR_EL1;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = MPAMIDR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = MPAMIDR_EL1;

```

## A.5.19 IMP\_CPUCFR\_EL1, CPU Configuration Register

This register provides configuration information for the core.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-77: AArch64\_imp\_cpucfr\_el1 bit assignments**

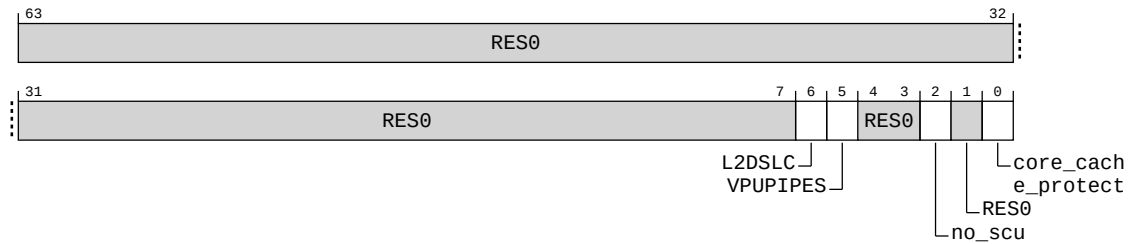


**Table A-217: IMP\_CPUCFR\_EL1 bit descriptions**

| Bits   | Name               | Description                                                                                                                                                                              | Reset |
|--------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:7] | RES0               | Reserved                                                                                                                                                                                 | RES0  |
| [6]    | L2DSLC             | L2 data RAM register slice presence. Possible values of this bit are:<br><br><b>0b0</b><br>No L2 data RAM register slice present<br><br><b>0b1</b><br>L2 data RAM register slice present | x     |
| [5]    | VPUPIPS            | Indicates the number of Vector Processing Unit (VPU) pipes. Possible values of this bit are:<br><br><b>0b0</b><br>2 x 128-bit<br><br><b>0b1</b><br>4 x 128-bit                           | x     |
| [4:3]  | RES0               | Reserved                                                                                                                                                                                 | RES0  |
| [2]    | no_scu             | Indicates whether the SCU is present or not. Possible values of this bit are:<br><br><b>0b0</b><br>The SCU is present.<br><br><b>0b1</b><br>The SCU is not present.                      | x     |
| [1]    | RES0               | Reserved                                                                                                                                                                                 | RES0  |
| [0]    | core_cache_protect | Indicates whether ECC is present or not. Possible values of this field are:<br><br><b>0b0</b><br>ECC is not present.<br><br><b>0b1</b><br>ECC is present.                                | x     |

## Access

MRS &lt;Xt&gt;, S3\_0\_C15\_C0\_0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1111 | 0b0000 | 0b000 |

## Accessibility

MRS &lt;Xt&gt;, S3\_0\_C15\_C0\_0

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TIDCP == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = IMP_CPUCFR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = IMP_CPUCFR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = IMP_CPUCFR_EL1;

```

## A.5.20 CLIDR\_EL1, Cache Level ID Register

Identifies the type of cache, or caches, that are implemented at each level and can be managed using the architected cache maintenance instructions that operate by set/way, up to a maximum of seven levels. Also identifies the Level of Coherence (LoC) and Level of Unification (LoU) for the cache hierarchy.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | x000 | 0000 | 0xxx | xxx0 | 1100 | 0011 | 0000 | 0000 | 0000 | 0001 | 0010 | 0011 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-78: AArch64\_clidr\_el1 bit assignments**

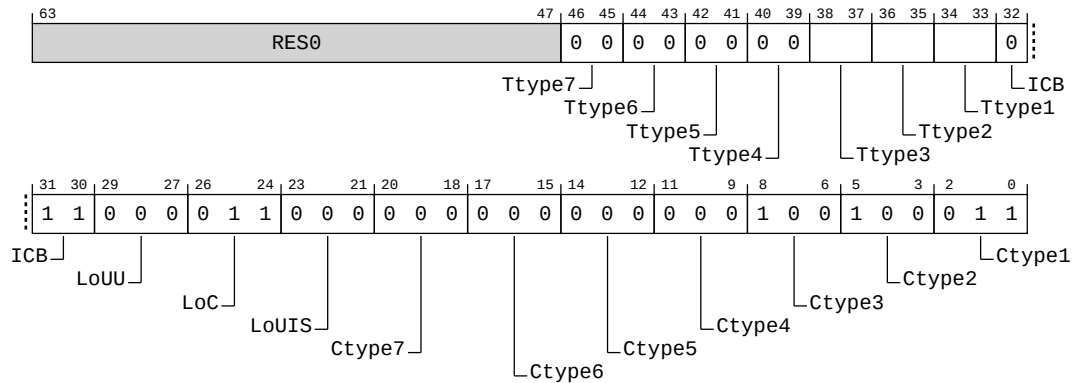


**Table A-219: CLIDR\_EL1 bit descriptions**

| Bits    | Name   | Description                                                                                                                                                                                                                                                                        | Reset |
|---------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:47] | RES0   | Reserved                                                                                                                                                                                                                                                                           | RES0  |
| [46:45] | Ttype7 | Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.<br><br><b>0b00</b><br>No Tag Cache. | 0b00  |
| [44:43] | Ttype6 | Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.<br><br><b>0b00</b><br>No Tag Cache. | 0b00  |
| [42:41] | Ttype5 | Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.<br><br><b>0b00</b><br>No Tag Cache. | 0b00  |
| [40:39] | Ttype4 | Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.<br><br><b>0b00</b><br>No Tag Cache. | 0b00  |

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|---------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [38:37] | Ttype3 | <p>Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.</p> <p><b>0b00</b><br/>No Tag Cache. This value is reported if the BROADCASTMTE pin is low.</p> <p><b>0b10</b><br/>Unified Allocation Tag and Data cache, Allocation Tags and Data in unified lines. This value is reported if the BROADCASTMTE pin is high.</p>                                                                                                                                  | xx    |
| [36:35] | Ttype2 | <p>Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.</p> <p><b>0b00</b><br/>No Tag Cache. This value is reported if the BROADCASTMTE pin is low.</p> <p><b>0b10</b><br/>Unified Allocation Tag and Data cache, Allocation Tags and Data in unified lines. This value is reported if the BROADCASTMTE pin is high.</p>                                                                                                                                  | xx    |
| [34:33] | Ttype1 | <p>Tag cache type. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy.</p> <p><b>0b00</b><br/>No Tag Cache. This value is reported if the BROADCASTMTE pin is low.</p> <p><b>0b10</b><br/>Unified Allocation Tag and Data cache, Allocation Tags and Data in unified lines. This value is reported if the BROADCASTMTE pin is high.</p>                                                                                                                                  | xx    |
| [32:30] | ICB    | <p>Inner cache boundary. This field indicates the boundary for caching Inner Cacheable memory regions.</p> <p><b>0b011</b><br/>L3 cache is the highest Inner Cacheable level.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0b011 |
| [29:27] | LoUU   | <p>Level of Unification Uniprocessor for the cache hierarchy.</p> <p>For a description of the values of this field, see Terminology for Clean, Invalidate, and Clean and Invalidate instructions.</p> <p><b>Note:</b><br/>This field does not describe the requirements for instruction cache invalidation. See AArch64-CTR_EL0.DIC.</p> <p><b>Note:</b><br/>When FEAT_S2FWB is implemented, the architecture requires that this field is zero so that no levels of data cache need to be cleaned in order to manage coherency with instruction fetches.</p> <p><b>0b000</b><br/>Level of Unification Uniprocessor is before the L1 data cache.</p> | 0b000 |
| [26:24] | LoC    | <p>Level of Coherence for the cache hierarchy.</p> <p>For a description of the values of this field, see Terminology for Clean, Invalidate, and Clean and Invalidate instructions.</p> <p><b>0b011</b><br/>Level 3</p>                                                                                                                                                                                                                                                                                                                                                                                                                              | 0b011 |



| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|---------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [23:21] | LoUIS  | <p>Level of Unification Inner Shareable for the cache hierarchy.</p> <p>For a description of the values of this field, see Terminology for Clean, Invalidate, and Clean and Invalidate instructions.</p> <p><b>Note:</b><br/>This field does not describe the requirements for instruction cache invalidation. See AArch64-CTR_EL0.DIC.</p> <p><b>Note:</b><br/>When FEAT_S2FWB is implemented, the architecture requires that this field is zero so that no levels of data cache need to be cleaned in order to manage coherency with instruction fetches.</p> <p><b>0b000</b><br/>No cache level needs cleaning to Point of Unification</p> | 0b000 |
| [20:18] | Ctype7 | <p>Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:</p> <p><b>0b000</b><br/>No cache.</p>                                                                                                                                                                                                                                                                                                                 | 0b000 |
| [17:15] | Ctype6 | <p>Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:</p> <p><b>0b000</b><br/>No cache.</p>                                                                                                                                                                                                                                                                                                                 | 0b000 |
| [14:12] | Ctype5 | <p>Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:</p> <p><b>0b000</b><br/>No cache.</p>                                                                                                                                                                                                                                                                                                                 | 0b000 |
| [11:9]  | Ctype4 | <p>Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:</p> <p><b>0b000</b><br/>No cache.</p>                                                                                                                                                                                                                                                                                                                 | 0b000 |
| [8:6]   | Ctype3 | <p>Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:</p> <p><b>0b100</b><br/>Unified instruction and data caches at L3</p>                                                                                                                                                                                                                                                                                 | 0b100 |
| [5:3]   | Ctype2 | <p>Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:</p> <p><b>0b100</b><br/>Unified instruction and data caches at L2</p>                                                                                                                                                                                                                                                                                 | 0b100 |

| Bits  | Name   | Description                                                                                                                                                                                                                                                                                                                                            | Reset |
|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [2:0] | Ctype1 | Cache Type fields. Indicate the type of cache that is implemented and can be managed using the architected cache maintenance instructions that operate by set/way at each level, from Level 1 up to a maximum of seven levels of cache hierarchy. Possible values of each field are:<br><br><b>0b011</b><br>Separate instruction and data caches at L1 | 0b011 |

### Access

MRS &lt;Xt&gt;, CLIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b001 | 0b0000 | 0b0000 | 0b001 |

### Accessibility

MRS &lt;Xt&gt;, CLIDR\_EL1

```

if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID2 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.TID4 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.CLIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = CLIDR_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = CLIDR_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = CLIDR_EL1;

```

## A.5.21 GMID\_EL1, Multiple tag transfer ID Register

Indicates the block size that is accessed by the LDGM and STGM System instructions.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

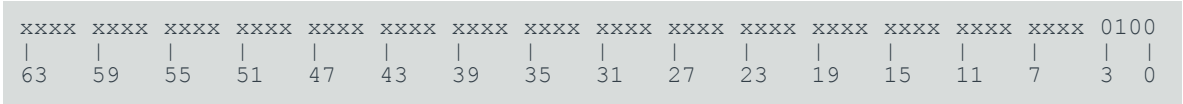
#### Functional group

Identification registers

#### Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-79: AArch64\_gmid\_el1 bit assignments

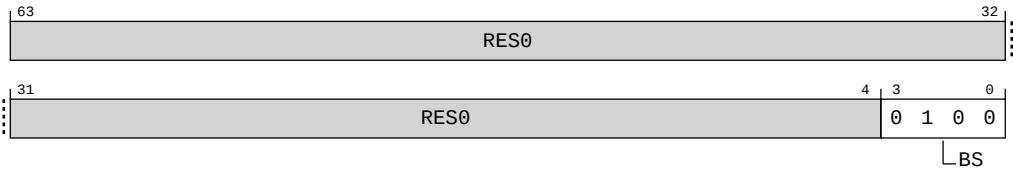


Table A-221: GMID\_EL1 bit descriptions

| Bits   | Name | Description                                                                                                                                                                        | Reset  |
|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:4] | RES0 | Reserved                                                                                                                                                                           | RES0   |
| [3:0]  | BS   | Log <sub>2</sub> of the block size in words. The minimum supported size is 16B (value == 2) and the maximum is 256B (value == 6).<br><br>0b0100<br><br>Log2 of the block size is 4 | 0b0100 |

Access

MRS <Xt>, GMID\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b001 | 0b0000 | 0b0000 | 0b100 |

Accessibility

MRS <Xt>, GMID\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID5 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = GMID_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = GMID_EL1;
```

```
elseif PSTATE.EL == EL3 then
    X[t, 64] = GMID_EL1;
```

A.5.22 CTR\_EL0, Cache Type Register

Provides information about the architecture of the caches.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Identification registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xx00 | 0100 | xx01 | 0100 | 0100 | 0100 | 11xx | xxxx | xxxx | 0100 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-80: AArch64\_ctr\_el0 bit assignments

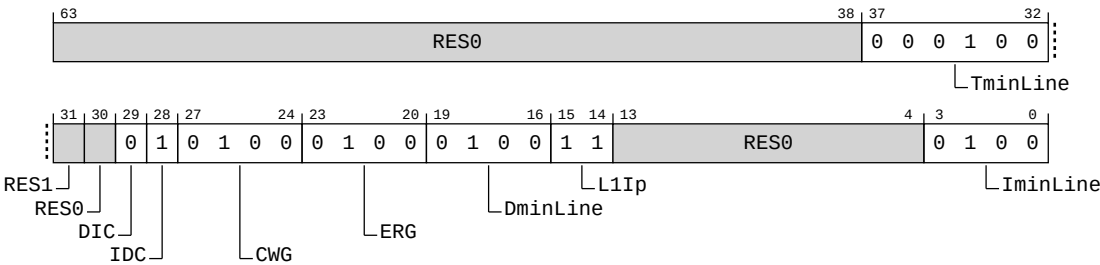


Table A-223: CTR\_EL0 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:38] | RES0 | Reserved    | RES0  |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                             | Reset    |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [37:32] | TminLine | Tag minimum Line. Log <sub>2</sub> of the number of words covered by Allocation Tags in the smallest cache line of all caches which can contain Allocation tags that are controlled by the PE.<br><br><b>0b000100</b><br>Log2 of number of words (64/4=16) covered by Allocation Tags in the smallest cache line of all caches          | 0b000100 |
| [31]    | RES1     | Reserved                                                                                                                                                                                                                                                                                                                                | RES1     |
| [30]    | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                | RES0     |
| [29]    | DIC      | Instruction cache invalidation requirements for data to instruction coherence.<br><br><b>0b0</b><br>Instruction cache invalidation to the point of unification is required for instruction to data coherence.                                                                                                                           | 0b0      |
| [28]    | IDC      | Data cache clean requirements for instruction to data coherence. The meaning of this bit is:<br><br><b>0b1</b><br>Data cache clean to the Point of Unification is not required for instruction to data coherence.                                                                                                                       | 0b1      |
| [27:24] | CWG      | Cache writeback granule. Log2 of the number of words of the maximum size of memory that can be overwritten as a result of the eviction of a cache entry that has had a memory location in it modified.<br><br><b>0b0100</b><br>64 bytes.                                                                                                | 0b0100   |
| [23:20] | ERG      | Exclusives reservation granule, and, if TME is implemented, transactional reservation granule. Log2 of the number of words of the maximum size of the reservation granule for the Load-Exclusive and Store-Exclusive instructions, and, if TME is implemented, for detecting transactional conflicts.<br><br><b>0b0100</b><br>64 bytes. | 0b0100   |
| [19:16] | DminLine | Log <sub>2</sub> of the number of words in the smallest cache line of all the data caches and unified caches that are controlled by the PE.<br><br><b>0b0100</b><br>64 bytes.                                                                                                                                                           | 0b0100   |
| [15:14] | L1lp     | Level 1 instruction cache policy. Indicates the indexing and tagging policy for the L1 instruction cache. Possible values of this field are:<br><br><b>0b11</b><br>Physical Index, Physical Tag (PIPT).                                                                                                                                 | 0b11     |
| [13:4]  | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                | RES0     |
| [3:0]   | IminLine | Log <sub>2</sub> of the number of words in the smallest cache line of all the instruction caches that are controlled by the PE.<br><br><b>0b0100</b><br>64 bytes.                                                                                                                                                                       | 0b0100   |

## Access

MRS <Xt>, CTR\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b0000 | 0b0000 | 0b001 |

## Accessibility

MRS <Xt>, CTR\_ELO

```

if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11') && SCTLR_EL1.UCT == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && HCR_EL2.TID2 == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HFGTR_EL2.CTR_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11' && SCTLR_EL2.UCT == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            X[t, 64] = CTR_ELO;
    elsif PSTATE.EL == EL1 then
        if EL2Enabled() && HCR_EL2.TID2 == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.CTR_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            X[t, 64] = CTR_ELO;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = CTR_ELO;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = CTR_ELO;

```

### A.5.23 DCZID\_ELO, Data Cache Zero ID Register

Indicates the block size that is written with byte values of 0 by the DC ZVA (Data Cache Zero by Address) System instruction.

If FEAT\_MTE is implemented, this register also indicates the granularity at which the DC GVA and DC GZVA instructions write.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Functional group

Identification registers

### Access type

See bit descriptions

### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | 0100 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-81: AArch64\_dcqid\_el0 bit assignments

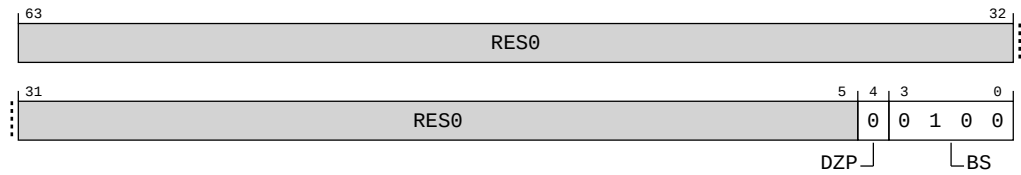


Table A-225: DCZID\_EL0 bit descriptions

| Bits   | Name | Description                                                                                                                                                                                                                                                                                             | Reset  |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:5] | RES0 | Reserved                                                                                                                                                                                                                                                                                                | RES0   |
| [4]    | DZP  | Data Zero Prohibited. This field indicates whether use of DC ZVA instructions is permitted or prohibited.<br><br>If FEAT_MTE is implemented, this field also indicates whether use of the DC GVA and DC GZVA instructions are permitted or prohibited.<br><br><b>0b0</b><br>Instructions are permitted. | 0b0    |
| [3:0]  | BS   | Log <sub>2</sub> of the block size in words. The maximum size supported is 2KB, indicated by value 0b1001.<br><br>If FEAT_MTE2 is implemented, the minimum size supported is 16 bytes, indicated by value 0b0010.<br><br><b>0b0100</b><br>Log <sub>2</sub> of the block size is 4                       | 0b0100 |

Access

MRS <Xt>, DCZID\_EL0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b0000 | 0b0000 | 0b111 |

Accessibility

MRS <Xt>, DCZID\_EL0

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
        HFGTR_EL2.DCZID_EL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = DCZID_EL0;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.DCZID_EL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
```

```

else
    X[t, 64] = DCZID_EL0;
elseif PSTATE.EL == EL2 then
    X[t, 64] = DCZID_EL0;
elseif PSTATE.EL == EL3 then
    X[t, 64] = DCZID_EL0;

```

## A.6 AArch64 Special-purpose registers summary

The following summary table provides an overview of all Special-purpose registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-227: Special-purpose registers summary**

| Name      | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                    |
|-----------|-----|-----|-----|-----|-----|----------------------------|--------|------------------------------------------------|
| SPSR_EL1  | 3   | 0   | C4  | C0  | 0   | See individual bit resets. | 64-bit | Saved Program Status Register (EL1)            |
| ELR_EL1   | 3   | 0   | C4  | C0  | 1   | See individual bit resets. | 64-bit | Exception Link Register (EL1)                  |
| SP_EL0    | 3   | 0   | C4  | C1  | 0   | See individual bit resets. | 64-bit | Stack Pointer (EL0)                            |
| DSPSR_EL0 | 3   | 3   | C4  | C5  | 0   | See individual bit resets. | 64-bit | Debug Saved Program Status Register            |
| DLR_EL0   | 3   | 3   | C4  | C5  | 1   | See individual bit resets. | 64-bit | Debug Link Register                            |
| SPSR_EL2  | 3   | 4   | C4  | C0  | 0   | See individual bit resets. | 64-bit | Saved Program Status Register (EL2)            |
| ELR_EL2   | 3   | 4   | C4  | C0  | 1   | See individual bit resets. | 64-bit | Exception Link Register (EL2)                  |
| SP_EL1    | 3   | 4   | C4  | C1  | 0   | See individual bit resets. | 64-bit | Stack Pointer (EL1)                            |
| SPSR_irq  | 3   | 4   | C4  | C3  | 0   | See individual bit resets. | 64-bit | Saved Program Status Register (IRQ mode)       |
| SPSR_abt  | 3   | 4   | C4  | C3  | 1   | See individual bit resets. | 64-bit | Saved Program Status Register (Abort mode)     |
| SPSR_und  | 3   | 4   | C4  | C3  | 2   | See individual bit resets. | 64-bit | Saved Program Status Register (Undefined mode) |
| SPSR_fiq  | 3   | 4   | C4  | C3  | 3   | See individual bit resets. | 64-bit | Saved Program Status Register (FIQ mode)       |
| SPSR_EL3  | 3   | 6   | C4  | C0  | 0   | See individual bit resets. | 64-bit | Saved Program Status Register (EL3)            |
| ELR_EL3   | 3   | 6   | C4  | C0  | 1   | See individual bit resets. | 64-bit | Exception Link Register (EL3)                  |
| SP_EL2    | 3   | 6   | C4  | C1  | 0   | See individual bit resets. | 64-bit | Stack Pointer (EL2)                            |



## A.7 AArch64 Performance Monitors registers summary

The following summary table provides an overview of all Performance Monitors registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-228: Performance Monitors registers summary**

| Name                        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                 |
|-----------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------------------|
| PMINTENSET_EL1              | 3   | 0   | C9  | C14 | 1   | See individual bit resets. | 64-bit | Performance Monitors Interrupt Enable Set Register          |
| PMINTENCLR_EL1              | 3   | 0   | C9  | C14 | 2   | See individual bit resets. | 64-bit | Performance Monitors Interrupt Enable Clear Register        |
| <a href="#">PMMIR_EL1</a>   | 3   | 0   | C9  | C14 | 6   | See individual bit resets. | 64-bit | Performance Monitors Machine Identification Register        |
| <a href="#">PMCR_ELO</a>    | 3   | 3   | C9  | C12 | 0   | See individual bit resets. | 64-bit | Performance Monitors Control Register                       |
| PMCNTENSET_ELO              | 3   | 3   | C9  | C12 | 1   | See individual bit resets. | 64-bit | Performance Monitors Count Enable Set Register              |
| PMCNTENCLR_ELO              | 3   | 3   | C9  | C12 | 2   | See individual bit resets. | 64-bit | Performance Monitors Count Enable Clear Register            |
| PMOVSCLR_ELO                | 3   | 3   | C9  | C12 | 3   | See individual bit resets. | 64-bit | Performance Monitors Overflow Flag Status Clear Register    |
| PMSWINC_ELO                 | 3   | 3   | C9  | C12 | 4   | See individual bit resets. | 64-bit | Performance Monitors Software Increment Register            |
| PMSELR_ELO                  | 3   | 3   | C9  | C12 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Counter Selection Register       |
| <a href="#">PMCEID0_ELO</a> | 3   | 3   | C9  | C12 | 6   | See individual bit resets. | 64-bit | Performance Monitors Common Event Identification Register 0 |
| <a href="#">PMCEID1_ELO</a> | 3   | 3   | C9  | C12 | 7   | See individual bit resets. | 64-bit | Performance Monitors Common Event Identification Register 1 |
| PMCCNTR_ELO                 | 3   | 3   | C9  | C13 | 0   | See individual bit resets. | 64-bit | Performance Monitors Cycle Count Register                   |
| PMXEVTYPER_ELO              | 3   | 3   | C9  | C13 | 1   | See individual bit resets. | 64-bit | Performance Monitors Selected Event Type Register           |
| PMXVCNTR_ELO                | 3   | 3   | C9  | C13 | 2   | See individual bit resets. | 64-bit | Performance Monitors Selected Event Count Register          |
| PMUSERENR_ELO               | 3   | 3   | C9  | C14 | 0   | See individual bit resets. | 64-bit | Performance Monitors User Enable Register                   |
| PMOVSSET_ELO                | 3   | 3   | C9  | C14 | 3   | See individual bit resets. | 64-bit | Performance Monitors Overflow Flag Status Set Register      |

| Name           | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| PMEVCNTR0_ELO  | 3   | 3   | C14 | C8  | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR1_ELO  | 3   | 3   | C14 | C8  | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR2_ELO  | 3   | 3   | C14 | C8  | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR3_ELO  | 3   | 3   | C14 | C8  | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR4_ELO  | 3   | 3   | C14 | C8  | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR5_ELO  | 3   | 3   | C14 | C8  | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR6_ELO  | 3   | 3   | C14 | C8  | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR7_ELO  | 3   | 3   | C14 | C8  | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR8_ELO  | 3   | 3   | C14 | C9  | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR9_ELO  | 3   | 3   | C14 | C9  | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR10_ELO | 3   | 3   | C14 | C9  | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR11_ELO | 3   | 3   | C14 | C9  | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR12_ELO | 3   | 3   | C14 | C9  | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR13_ELO | 3   | 3   | C14 | C9  | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR14_ELO | 3   | 3   | C14 | C9  | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR15_ELO | 3   | 3   | C14 | C9  | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR16_ELO | 3   | 3   | C14 | C10 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR17_ELO | 3   | 3   | C14 | C10 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR18_ELO | 3   | 3   | C14 | C10 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR19_ELO | 3   | 3   | C14 | C10 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR20_ELO | 3   | 3   | C14 | C10 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR21_ELO | 3   | 3   | C14 | C10 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR22_ELO | 3   | 3   | C14 | C10 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| PMEVCNTR23_ELO  | 3   | 3   | C14 | C10 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR24_ELO  | 3   | 3   | C14 | C11 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR25_ELO  | 3   | 3   | C14 | C11 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR26_ELO  | 3   | 3   | C14 | C11 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR27_ELO  | 3   | 3   | C14 | C11 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR28_ELO  | 3   | 3   | C14 | C11 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR29_ELO  | 3   | 3   | C14 | C11 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVCNTR30_ELO  | 3   | 3   | C14 | C11 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| PMEVTYPER0_ELO  | 3   | 3   | C14 | C12 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER1_ELO  | 3   | 3   | C14 | C12 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER2_ELO  | 3   | 3   | C14 | C12 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER3_ELO  | 3   | 3   | C14 | C12 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER4_ELO  | 3   | 3   | C14 | C12 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER5_ELO  | 3   | 3   | C14 | C12 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER6_ELO  | 3   | 3   | C14 | C12 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER7_ELO  | 3   | 3   | C14 | C12 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER8_ELO  | 3   | 3   | C14 | C13 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER9_ELO  | 3   | 3   | C14 | C13 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER10_ELO | 3   | 3   | C14 | C13 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER11_ELO | 3   | 3   | C14 | C13 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER12_ELO | 3   | 3   | C14 | C13 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER13_ELO | 3   | 3   | C14 | C13 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |
| PMEVTYPER14_ELO | 3   | 3   | C14 | C13 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers  |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                      |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------|
| PMEVTYPER15_ELO | 3   | 3   | C14 | C13 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER16_ELO | 3   | 3   | C14 | C14 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER17_ELO | 3   | 3   | C14 | C14 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER18_ELO | 3   | 3   | C14 | C14 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER19_ELO | 3   | 3   | C14 | C14 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER20_ELO | 3   | 3   | C14 | C14 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER21_ELO | 3   | 3   | C14 | C14 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER22_ELO | 3   | 3   | C14 | C14 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER23_ELO | 3   | 3   | C14 | C14 | 7   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER24_ELO | 3   | 3   | C14 | C15 | 0   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER25_ELO | 3   | 3   | C14 | C15 | 1   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER26_ELO | 3   | 3   | C14 | C15 | 2   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER27_ELO | 3   | 3   | C14 | C15 | 3   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER28_ELO | 3   | 3   | C14 | C15 | 4   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER29_ELO | 3   | 3   | C14 | C15 | 5   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMEVTYPER30_ELO | 3   | 3   | C14 | C15 | 6   | See individual bit resets. | 64-bit | Performance Monitors Event Type Registers        |
| PMCCFILTR_ELO   | 3   | 3   | C14 | C15 | 7   | See individual bit resets. | 64-bit | Performance Monitors Cycle Count Filter Register |

## A.7.1 PMMIR\_EL1, Performance Monitors Machine Identification Register

Describes Performance Monitors parameters specific to the implementation to software.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

Functional group

Performance Monitors registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 | 0000 | 1010 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-82: AArch64\_pmmir\_el1 bit assignments

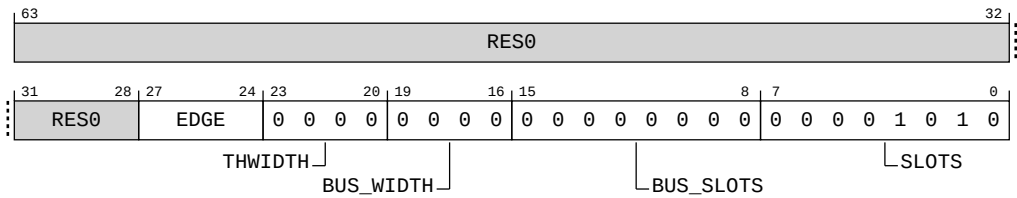


Table A-229: PMMIR\_EL1 bit descriptions

| Bits    | Name      | Description                                                                                                                                                                                                                 | Reset                                                                           |
|---------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [63:28] | RES0      | Reserved                                                                                                                                                                                                                    | RES0                                                                            |
| [27:24] | EDGE      | PMU event edge detection. Indicates implementation of the FEAT_PMUv3_EDGE feature.<br><br><b>0b0000</b><br>FEAT_PMUv3_EDGE is not implemented.<br><br><b>0b0001</b><br>FEAT_PMUv3_EDGE is implemented.                      | The reset values can be the following: 0b0000, 0b0001, respective to the value. |
| [23:20] | THWIDTH   | AArch64-PMEVTYPER<n>_ELO.TH width. Indicates implementation of the FEAT_PMUv3_TH feature, and, if implemented, the size of the AArch64-PMEVTYPER<n>_ELO.TH field.<br><br><b>0b0000</b><br>FEAT_PMUv3_TH is not implemented. | 0b0000                                                                          |
| [19:16] | BUS_WIDTH | Bus width. Indicates the number of bytes each BUS_ACCESS event relates to. Encoded as Log <sub>2</sub> (number of bytes), plus one.<br><br><b>0b0000</b><br>The information is not available.                               | 0b0000                                                                          |

| Bits   | Name      | Description                                                                                                                                                                                                                                                                                      | Reset |
|--------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15:8] | BUS_SLOTS | Bus count. The largest value by which the BUS_ACCESS event might increment in a single BUS_CYCLES cycle.<br><br><b>0b00000000</b><br>The largest value by which the BUS_ACCESS event might increment in a single BUS_CYCLES cycle is 0                                                           | 0x00  |
| [7:0]  | SLOTS     | Operation width. The largest value by which the STALL_SLOT event might increment in a single cycle. If the STALL_SLOT event is not implemented, this field might read as zero.<br><br><b>0b00001010</b><br>The largest value by which the STALL_SLOT PMU event may increment in one cycle is 10. | 0x0A  |

## Access

MRS <Xt>, PMMIR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1110 | 0b110 |

## Accessibility

MRS <Xt>, PMMIR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.PMMIR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMMIR_EL1;
    elseif PSTATE.EL == EL2 then
        if MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMMIR_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = PMMIR_EL1;

```

A.7.2 PMCR\_EL0, Performance Monitors Control Register

Provides details of the Performance Monitors implementation, including the number of counters implemented, and configures and controls the counters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Performance Monitors registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | xxxx | xxxx | xxxx | xxxx | xxx0 | x000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-83: AArch64\_pmcr\_el0 bit assignments

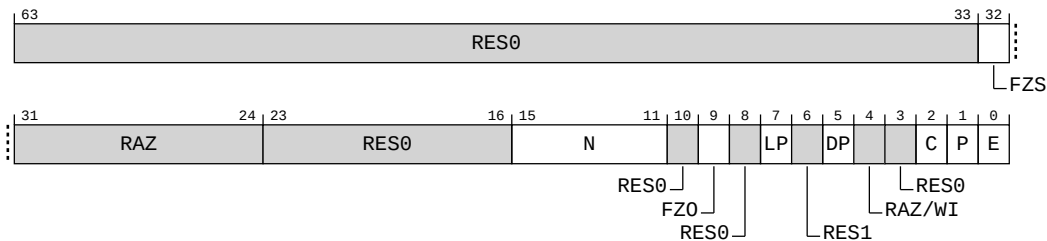


Table A-231: PMCR\_EL0 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:33] | RES0 | Reserved    | RES0  |

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset                                                                             |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| [32]    | FZS  | Freeze-on-SPE event.<br><br>Stop counters when AArch64-PMBLIMITR_EL1.{PMFZ,E} == {1,1} and AArch64-PMBSR_EL1.S == 1.<br><br>In the description of this field: <ul style="list-style-type: none"> <li>If EL2 is implemented, then PMN is AArch64-MDCR_EL2.HPMN.</li> <li>If EL2 is not implemented, then PMN is PMCR_ELO.N.</li> </ul> <b>0b0</b><br>Do not freeze on a Statistical Profiling Buffer Management event.                      | x                                                                                 |
| [31:24] | RAZ  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                   | RAZ                                                                               |
| [23:16] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES0                                                                              |
| [15:11] | N    | Number of event counters:<br><br><b>0b00110</b><br>When configured for 6 PMU counters, denotes 6 PMU counters implemented<br><br><b>0b11111</b><br>When configured for 31 PMU counters, denotes 31 PMU counters implemented                                                                                                                                                                                                                | The reset values can be the following: 0b00110, 0b11111, respective to the value. |
| [10]    | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES0                                                                              |
| [9]     | FZO  | Freeze-on-overflow.<br><br>Stop event counters on overflow.<br><br>In the description of this field: <ul style="list-style-type: none"> <li>If EL2 is implemented, then PMN is AArch64-MDCR_EL2.HPMN.</li> <li>If EL2 is not implemented, then PMN is PMCR_ELO.N.</li> </ul> <b>0b0</b><br>Do not freeze on overflow.                                                                                                                      | x                                                                                 |
| [8]     | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES0                                                                              |
| [7]     | LP   | Long event counter enable.<br><br>Determines which event counter bit generates an overflow recorded by AArch32-PMOVSr[n].<br><br>In the description of this field: <ul style="list-style-type: none"> <li>If EL2 is implemented, then PMN is AArch64-MDCR_EL2.HPMN.</li> <li>If EL2 is not implemented, then PMN is PMCR_ELO.N.</li> </ul> <b>0b0</b><br>Affected counters overflow on unsigned overflow of AArch64-PMEVCNTR<n>_ELO[31:0]. | x                                                                                 |
| [6]     | RES1 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES1                                                                              |



| Bits | Name          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset         |
|------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [5]  | DP            | <p>Disable cycle counter when event counting is prohibited.</p> <p><b>0b0</b></p> <p>Cycle counting by AArch64-PMCCNTR_ELO is not affected by this mechanism.</p> <p><b>0b1</b></p> <p>Cycle counting by AArch64-PMCCNTR_ELO is disabled in prohibited regions and when event counting is frozen:</p> <ul style="list-style-type: none"> <li>• If FEAT_PMUv3p1 is implemented, EL2 is implemented, and AArch64-MDCR_EL2.HPMD is 1, then cycle counting by AArch64-PMCCNTR_ELO is disabled at EL2.</li> <li>• If FEAT_PMUv3p7 is implemented, EL3 is implemented and using AArch64, and AArch64-MDCR_EL3.MPMX is 1, then cycle counting by AArch64-PMCCNTR_ELO is disabled at EL3.</li> <li>• If FEAT_PMUv3p7 is implemented and event counting is frozen by PMCR_ELO.FZO, then cycle counting by AArch64-PMCCNTR_ELO is disabled.</li> <li>• If FEAT_SPE_DPFZS is implemented and event counting is frozen by PMCR_ELO.FZS, then cycle counting by AArch64-PMCCNTR_ELO is disabled.</li> <li>• If FEAT_PMUv3p7 is implemented, FEAT_SPE_DPFZS is not implemented, and event counting is frozen by PMCR_ELO.FZS, then it is IMPLEMENTATION DEFINED whether cycle counting by AArch64-PMCCNTR_ELO is disabled.</li> <li>• If EL3 is implemented, AArch64-MDCR_EL3.SPME is 0, and either FEAT_PMUv3p7 is not implemented or AArch64-MDCR_EL3.MPMX is 0, then cycle counting by AArch64-PMCCNTR_ELO is disabled at EL3 and in Secure state.</li> </ul> | x             |
| [4]  | <b>RAZ/WI</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>RAZ/WI</b> |
| [3]  | <b>RES0</b>   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>RES0</b>   |
| [2]  | C             | <p>Cycle counter reset. The effects of writing to this bit are:</p> <p><b>0b0</b></p> <p>No action.</p> <p><b>0b1</b></p> <p>Reset AArch64-PMCCNTR_ELO to zero.</p> <p>Access to this field is: WO/<b>RAZ</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0b0           |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [1]  | P    | <p>Event counter reset.</p> <p>In the description of this field:</p> <ul style="list-style-type: none"> <li>If EL2 is implemented and is using AArch64, PMN is AArch64-MDCR_EL2.HPMN.</li> </ul> <p><b>0b0</b></p> <p>No action.</p> <p><b>0b1</b></p> <p>If n is in the range of affected event counters, resets each event counter AArch64-PMEVCNTR&lt;n&gt;_ELO to zero.</p> <p>Access to this field is: WO/<b>RAZ</b></p> | 0b0   |
| [0]  | E    | <p>Enable.</p> <p>In the description of this field:</p> <ul style="list-style-type: none"> <li>If EL2 is implemented, then PMN is AArch64-MDCR_EL2.HPMN.</li> <li>If EL2 is not implemented, then PMN is PMCR_ELO.N.</li> </ul> <p><b>0b0</b></p> <p>Affected counters are disabled and do not count.</p> <p><b>0b1</b></p> <p>Affected counters are enabled by AArch64-PMCNTENSET_ELO.</p>                                   | 0b0   |

### Access

MRS <Xt>, PMCR\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1001 | 0b1100 | 0b000 |

MSR PMCR\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1001 | 0b1100 | 0b000 |

### Accessibility

MRS <Xt>, PMCR\_ELO

```

if PSTATE.EL == EL0 then
    if PMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && MDCR_EL2.TPMCR == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif MDCR_EL3.TPM == '1' then

```

```

        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMCR_EL0;
    elseif PSTATE.EL == EL1 then
        if EL2Enabled() && MDCR_EL2.TPM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif EL2Enabled() && MDCR_EL2.TPMCR == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMCR_EL0;
    elseif PSTATE.EL == EL2 then
        if MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMCR_EL0;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = PMCR_EL0;

```

## MSR PMCR\_EL0, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    if PMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HDFGWTR_EL2.PMCR_EL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPMCR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        PMCR_EL0 = X[t, 64];
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGWTR_EL2.PMCR_EL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPMCR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        PMCR_EL0 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then

```

```

        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        PMCR_ELO = X[t, 64];
elseif PSTATE.EL == EL3 then
    PMCR_ELO = X[t, 64];

```

### A.7.3 PMCEID0\_ELO, Performance Monitors Common Event Identification Register 0

Defines which Common architectural events and Common microarchitectural events are implemented, or counted, using PMU events in the ranges 0x0000 to 0x001F and 0x4000 to 0x401F.

For more information about the Common events and the use of the PMCEID<n>\_ELO registers see *The PMU event number space and common events* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Performance Monitors registers

##### Access type

See bit descriptions

##### Reset value

0000 1111 0000 1111 0001 1010 0111 1111 0111 1111 1111 1111 0110 1111 0011  
1111

## Bit descriptions

Figure A-84: AArch64\_pmceid0\_el0 bit assignments

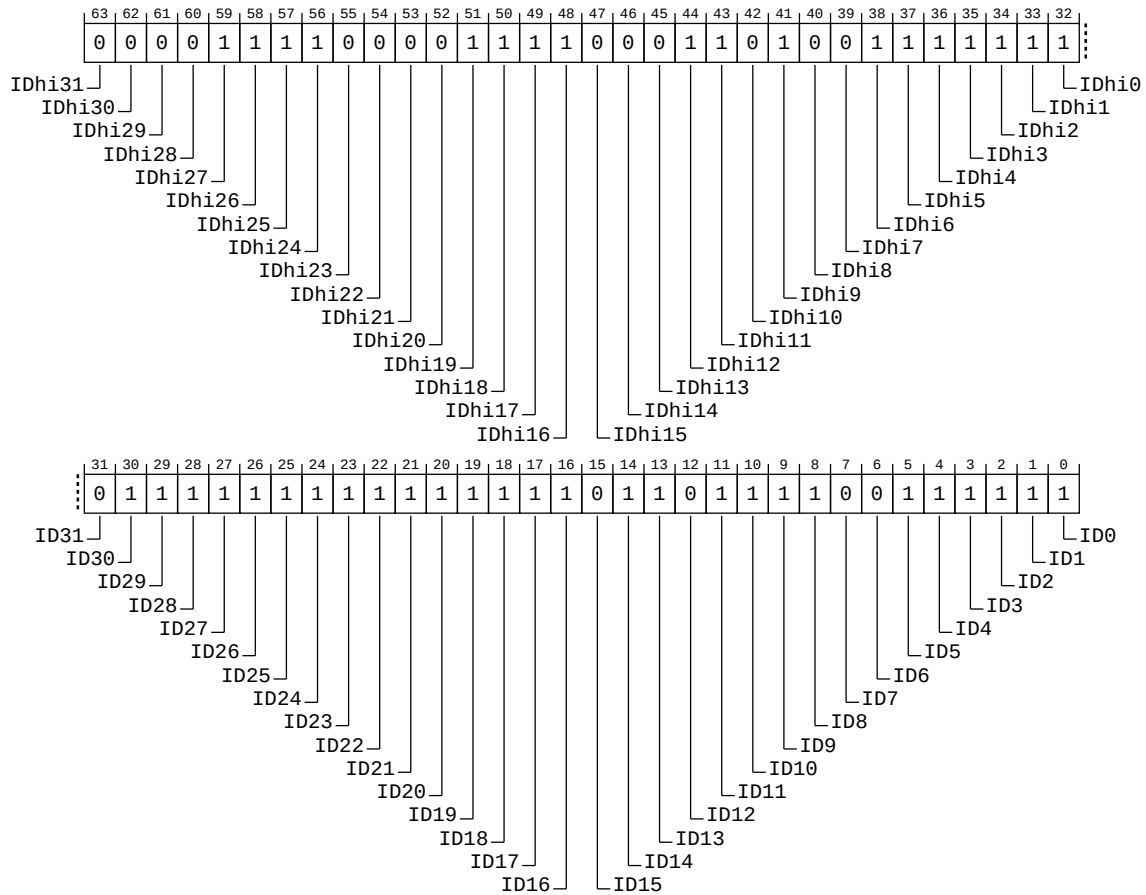


Table A-234: PMCEID0\_ELO bit descriptions

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [63] | IDhi31 | IDhi31 corresponds to a Reserved Event event (0x401f)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [62] | IDhi30 | IDhi30 corresponds to a Reserved Event event (0x401e)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [61] | IDhi29 | IDhi29 corresponds to a Reserved Event event (0x401d)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [60] | IDhi28 | IDhi28 corresponds to a Reserved Event event (0x401c)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [59] | IDHi27 | IDHi27 corresponds to common event (0x401b) CTI_TRIGOUT7<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [58] | IDHi26 | IDHi26 corresponds to common event (0x401a) CTI_TRIGOUT6<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [57] | IDHi25 | IDHi25 corresponds to common event (0x4019) CTI_TRIGOUT5<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [56] | IDHi24 | IDHi24 corresponds to common event (0x4018) CTI_TRIGOUT4<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [55] | IDHi23 | IDHi23 corresponds to a Reserved Event event (0x4017)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [54] | IDHi22 | IDHi22 corresponds to a Reserved Event event (0x4016)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [53] | IDHi21 | IDHi21 corresponds to a Reserved Event event (0x4015)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [52] | IDHi20 | IDHi20 corresponds to a Reserved Event event (0x4014)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [51] | IDHi19 | IDHi19 corresponds to common event (0x4013) TRCEXTOUT3<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [50] | IDHi18 | IDHi18 corresponds to common event (0x4012) TRCEXTOUT2<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [49] | IDHi17 | IDHi17 corresponds to common event (0x4011) TRCEXTOUT1<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [48] | IDHi16 | IDHi16 corresponds to common event (0x4010) TRCEXTOUT0<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [47] | IDHi15 | IDHi15 corresponds to common event (0x400f) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |
| [46] | IDHi14 | IDHi14 corresponds to common event (0x400e) TRB_TRIG<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |

| Bits | Name   | Description                                                                                                                       | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------|-------|
| [45] | IDhi13 | IDhi13 corresponds to common event (0x400d) PMU_OVFS<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [44] | IDhi12 | IDhi12 corresponds to common event (0x400c) TRB_WRAP<br><b>0b1</b><br>The Common event is implemented.                            | 0b1   |
| [43] | IDhi11 | IDhi11 corresponds to common event (0x400b) L3D_CACHE_LMISS_RD<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [42] | IDhi10 | IDhi10 corresponds to common event (0x400a) L2I_CACHE_LMISS<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [41] | IDhi9  | IDhi9 corresponds to common event (0x4009) L2D_CACHE_LMISS_RD<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [40] | IDhi8  | IDhi8 corresponds to common event (0x4008) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.         | 0b0   |
| [39] | IDhi7  | IDhi7 corresponds to common event (0x4007) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.         | 0b0   |
| [38] | IDhi6  | IDhi6 corresponds to common event (0x4006) L1I_CACHE_LMISS<br><b>0b1</b><br>The Common event is implemented.                      | 0b1   |
| [37] | IDhi5  | IDhi5 corresponds to common event (0x4005) STALL_BACKEND_MEM<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [36] | IDhi4  | IDhi4 corresponds to common event (0x4004) CNT_CYCLES<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [35] | IDhi3  | IDhi3 corresponds to common event (0x4003) SAMPLE_COLLISION<br><b>0b1</b><br>The Common event is implemented.                     | 0b1   |
| [34] | IDhi2  | IDhi2 corresponds to common event (0x4002) SAMPLE_FILTRATE<br><b>0b1</b><br>The Common event is implemented.                      | 0b1   |
| [33] | IDhi1  | IDhi1 corresponds to common event (0x4001) SAMPLE_FEED<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [32] | IDhi0  | IDhi0 corresponds to common event (0x4000) SAMPLE_POP<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |

| Bits | Name | Description                                                                                                                      | Reset |
|------|------|----------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | ID31 | ID31 corresponds to common event (0x1f) L1D_CACHE_ALLOCATE<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [30] | ID30 | ID30 corresponds to common event (0x1e) CHAIN<br><b>0b1</b><br>The Common event is implemented.                                  | 0b1   |
| [29] | ID29 | ID29 corresponds to common event (0x1d) BUS_CYCLES<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [28] | ID28 | ID28 corresponds to common event (0x1c) TTBR_WRITE_RETIRED<br><b>0b1</b><br>The Common event is implemented.                     | 0b1   |
| [27] | ID27 | ID27 corresponds to common event (0x1b) INST_SPEC<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [26] | ID26 | ID26 corresponds to common event (0x1a) MEMORY_ERROR<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [25] | ID25 | ID25 corresponds to common event (0x19) BUS_ACCESS<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [24] | ID24 | ID24 corresponds to common event (0x18) L2D_CACHE_WB<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [23] | ID23 | ID23 corresponds to common event (0x17) L2D_CACHE_REFILL<br><b>0b1</b><br>The Common event is implemented.                       | 0b1   |
| [22] | ID22 | ID22 corresponds to common event (0x16) L2D_CACHE<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [21] | ID21 | ID21 corresponds to common event (0x15) L1D_CACHE_WB<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [20] | ID20 | ID20 corresponds to common event (0x14) L1I_CACHE<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [19] | ID19 | ID19 corresponds to common event (0x13) MEM_ACCESS<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [18] | ID18 | ID18 corresponds to common event (0x12) BR_PRED<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |



| Bits | Name | Description                                                                                                                         | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------|-------|
| [17] | ID17 | ID17 corresponds to common event (0x11) CPU_CYCLES<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |
| [16] | ID16 | ID16 corresponds to common event (0x10) BR_MIS_PRED<br><b>0b1</b><br>The Common event is implemented.                               | 0b1   |
| [15] | ID15 | ID15 corresponds to common event (0xf) UNALIGNED_LDST_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [14] | ID14 | ID14 corresponds to common event (0xe) BR_RETURN_RETIRED<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [13] | ID13 | ID13 corresponds to common event (0xd) BR_IMMED_RETIRED<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [12] | ID12 | ID12 corresponds to common event (0xc) PC_WRITE_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted.       | 0b0   |
| [11] | ID11 | ID11 corresponds to common event (0xb) CID_WRITE_RETIRED<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [10] | ID10 | ID10 corresponds to common event (0xa) EXC_RETURN<br><b>0b1</b><br>The Common event is implemented.                                 | 0b1   |
| [9]  | ID9  | ID9 corresponds to common event (0x9) EXC_TAKEN<br><b>0b1</b><br>The Common event is implemented.                                   | 0b1   |
| [8]  | ID8  | ID8 corresponds to common event (0x8) INST_RETIRED<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |
| [7]  | ID7  | ID7 corresponds to common event (0x7) ST_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted.              | 0b0   |
| [6]  | ID6  | ID6 corresponds to common event (0x6) LD_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted.              | 0b0   |
| [5]  | ID5  | ID5 corresponds to common event (0x5) L1D_TLB_REFILL<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [4]  | ID4  | ID4 corresponds to common event (0x4) L1D_CACHE<br><b>0b1</b><br>The Common event is implemented.                                   | 0b1   |

| Bits | Name | Description                                                                                                  | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------|-------|
| [3]  | ID3  | ID3 corresponds to common event (0x3) L1D_CACHE_REFILL<br><br><b>0b1</b><br>The Common event is implemented. | 0b1   |
| [2]  | ID2  | ID2 corresponds to common event (0x2) L1I_TLB_REFILL<br><br><b>0b1</b><br>The Common event is implemented.   | 0b1   |
| [1]  | ID1  | ID1 corresponds to common event (0x1) L1I_CACHE_REFILL<br><br><b>0b1</b><br>The Common event is implemented. | 0b1   |
| [0]  | ID0  | ID0 corresponds to common event (0x0) SW_INCR<br><br><b>0b1</b><br>The Common event is implemented.          | 0b1   |

## Access

MRS &lt;Xt&gt;, PMCEID0\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1001 | 0b1100 | 0b110 |

## Accessibility

MRS &lt;Xt&gt;, PMCEID0\_ELO

```

if PSTATE.EL == EL0 then
    if PMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HDFGRTR_EL2.PMCEIDn_EL0 == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMCEID0_ELO;
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.PMCEIDn_EL0 == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif MDCR_EL3.TPM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMCEID0_ELO;
        elsif PSTATE.EL == EL2 then
            if MDCR_EL3.TPM == '1' then
                if Halted() && EDSCR.SDD == '1' then

```

```

        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMCEID0_EL0;
elseif PSTATE.EL == EL3 then
    X[t, 64] = PMCEID0_EL0;

```

## A.7.4 PMCEID1\_EL0, Performance Monitors Common Event Identification Register 1

Defines which Common architectural events and Common microarchitectural events are implemented, or counted, using PMU events in the ranges 0x0020 to 0x003F and 0x4020 to 0x403F.

For more information about the Common events and the use of the PMCEID<n>\_EL0 registers see *The PMU event number space and common events* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Performance Monitors registers

#### Access type

See bit descriptions

#### Reset value

0000 0000 0000 0000 0000 0000 0111 0111 1111 1110 1111 0010 1010 1110 0111  
1111

## Bit descriptions

Figure A-85: AArch64\_pmceid1\_el0 bit assignments

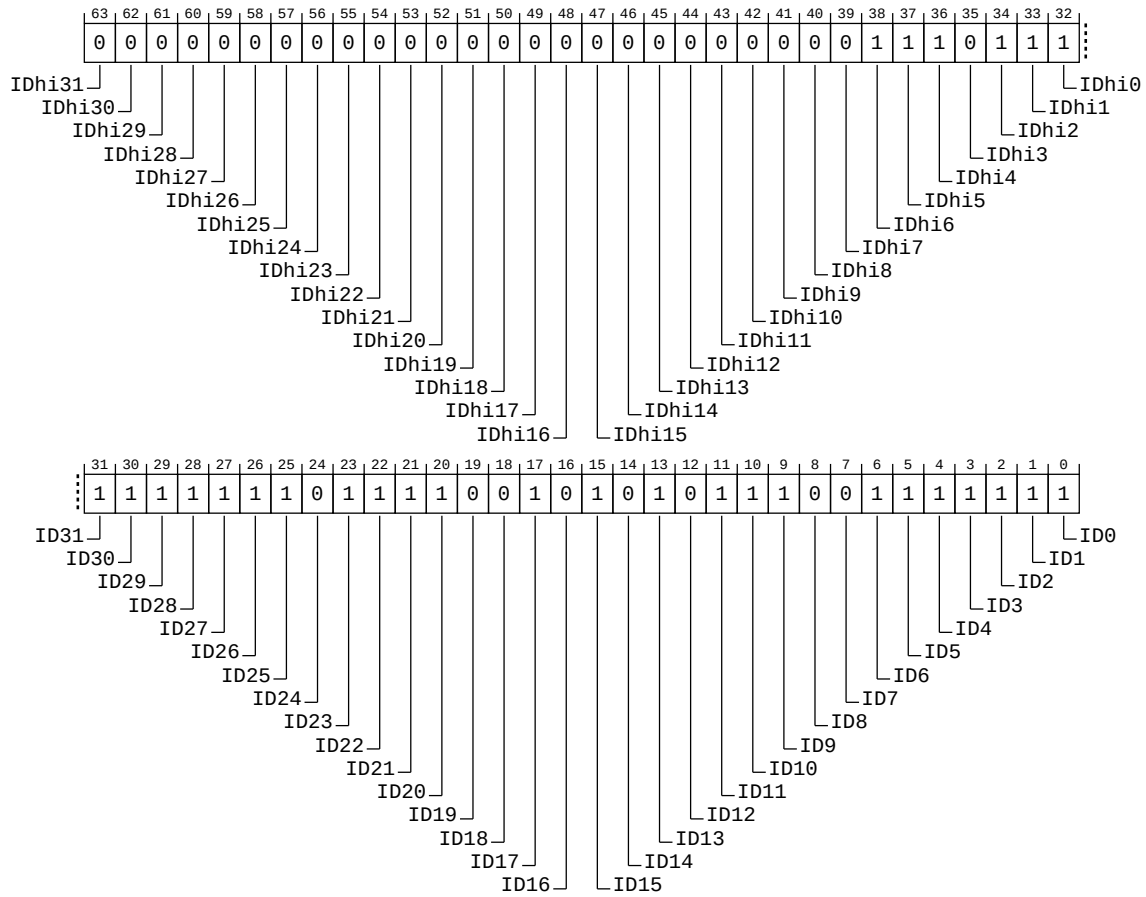


Table A-236: PMCEID1\_ELO bit descriptions

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [63] | IDhi31 | IDhi31 corresponds to a Reserved Event event (0x403f)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [62] | IDhi30 | IDhi30 corresponds to a Reserved Event event (0x403e)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [61] | IDhi29 | IDhi29 corresponds to a Reserved Event event (0x403d)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [60] | IDhi28 | IDhi28 corresponds to a Reserved Event event (0x403c)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [59] | IDHi27 | IDHi27 corresponds to a Reserved Event event (0x403b)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [58] | IDHi26 | IDHi26 corresponds to a Reserved Event event (0x403a)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [57] | IDHi25 | IDHi25 corresponds to a Reserved Event event (0x4039)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [56] | IDHi24 | IDHi24 corresponds to a Reserved Event event (0x4038)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [55] | IDHi23 | IDHi23 corresponds to a Reserved Event event (0x4037)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [54] | IDHi22 | IDHi22 corresponds to a Reserved Event event (0x4036)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [53] | IDHi21 | IDHi21 corresponds to a Reserved Event event (0x4035)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [52] | IDHi20 | IDHi20 corresponds to a Reserved Event event (0x4034)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [51] | IDHi19 | IDHi19 corresponds to a Reserved Event event (0x4033)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [50] | IDHi18 | IDHi18 corresponds to a Reserved Event event (0x4032)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [49] | IDHi17 | IDHi17 corresponds to a Reserved Event event (0x4031)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [48] | IDHi16 | IDHi16 corresponds to a Reserved Event event (0x4030)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [47] | IDHi15 | IDHi15 corresponds to a Reserved Event event (0x402f)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [46] | IDHi14 | IDHi14 corresponds to a Reserved Event event (0x402e)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [45] | IDHi13 | IDHi13 corresponds to a Reserved Event event (0x402d)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [44] | IDHi12 | IDHi12 corresponds to a Reserved Event event (0x402c)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [43] | IDHi11 | IDHi11 corresponds to a Reserved Event event (0x402b)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [42] | IDHi10 | IDHi10 corresponds to a Reserved Event event (0x402a)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [41] | IDHi9  | IDHi9 corresponds to a Reserved Event event (0x4029)<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |
| [40] | IDHi8  | IDHi8 corresponds to a Reserved Event event (0x4028)<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |
| [39] | IDHi7  | IDHi7 corresponds to a Reserved Event event (0x4027)<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |
| [38] | IDHi6  | IDHi6 corresponds to common event (0x4026) MEM_ACCESS_CHECKED_WR<br><b>0b1</b><br>The Common event is implemented.          | 0b1   |
| [37] | IDHi5  | IDHi5 corresponds to common event (0x4025) MEM_ACCESS_CHECKED_RD<br><b>0b1</b><br>The Common event is implemented.          | 0b1   |
| [36] | IDHi4  | IDHi4 corresponds to common event (0x4024) MEM_ACCESS_CHECKED<br><b>0b1</b><br>The Common event is implemented.             | 0b1   |
| [35] | IDHi3  | IDHi3 corresponds to common event (0x4023) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.   | 0b0   |
| [34] | IDHi2  | IDHi2 corresponds to common event (0x4022) ST_ALIGN_LAT<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [33] | IDHi1  | IDHi1 corresponds to common event (0x4021) LD_ALIGN_LAT<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [32] | IDHi0  | IDHi0 corresponds to common event (0x4020) LDST_ALIGN_LAT<br><b>0b1</b><br>The Common event is implemented.                 | 0b1   |

| Bits | Name | Description                                                                                                                    | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | ID31 | ID31 corresponds to common event (0x3f) STALL_SLOT<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [30] | ID30 | ID30 corresponds to common event (0x3e) STALL_SLOT_FRONTEND<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [29] | ID29 | ID29 corresponds to common event (0x3d) STALL_SLOT_BACKEND<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [28] | ID28 | ID28 corresponds to common event (0x3c) STALL<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |
| [27] | ID27 | ID27 corresponds to common event (0x3b) OP_SPEC<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [26] | ID26 | ID26 corresponds to common event (0x3a) OP_RETIRED<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [25] | ID25 | ID25 corresponds to common event (0x39) L1D_CACHE_LMISS_RD<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [24] | ID24 | ID24 corresponds to common event (0x38) REMOTE_ACCESS_RD<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [23] | ID23 | ID23 corresponds to common event (0x37) LL_CACHE_MISS_RD<br><b>0b1</b><br>The Common event is implemented.                     | 0b1   |
| [22] | ID22 | ID22 corresponds to common event (0x36) LL_CACHE_RD<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [21] | ID21 | ID21 corresponds to common event (0x35) ITLB_WALK<br><b>0b1</b><br>The Common event is implemented.                            | 0b1   |
| [20] | ID20 | ID20 corresponds to common event (0x34) DTLB_WALK<br><b>0b1</b><br>The Common event is implemented.                            | 0b1   |
| [19] | ID19 | ID19 corresponds to a Reserved Event event (0x33)<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [18] | ID18 | ID18 corresponds to a Reserved Event event (0x32)<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |

| Bits | Name | Description                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| [17] | ID17 | ID17 corresponds to common event (0x31) REMOTE_ACCESS<br><b>0b1</b><br>The Common event is implemented.                       | 0b1   |
| [16] | ID16 | ID16 corresponds to common event (0x30) L2I_TLB<br><b>0b0</b><br>The Common event is not implemented, or not counted.         | 0b0   |
| [15] | ID15 | ID15 corresponds to common event (0x2f) L2D_TLB<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [14] | ID14 | ID14 corresponds to common event (0x2e) L2I_TLB_REFILL<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |
| [13] | ID13 | ID13 corresponds to common event (0x2d) L2D_TLB_REFILL<br><b>0b1</b><br>The Common event is implemented.                      | 0b1   |
| [12] | ID12 | ID12 corresponds to common event (0x2c) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [11] | ID11 | ID11 corresponds to common event (0x2b) L3D_CACHE<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [10] | ID10 | ID10 corresponds to common event (0x2a) L3D_CACHE_REFILL<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [9]  | ID9  | ID9 corresponds to common event (0x29) L3D_CACHE_ALLOCATE<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [8]  | ID8  | ID8 corresponds to common event (0x28) L2I_CACHE_REFILL<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [7]  | ID7  | ID7 corresponds to common event (0x27) L2I_CACHE<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [6]  | ID6  | ID6 corresponds to common event (0x26) L1I_TLB<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [5]  | ID5  | ID5 corresponds to common event (0x25) L1D_TLB<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [4]  | ID4  | ID4 corresponds to common event (0x24) STALL_BACKEND<br><b>0b1</b><br>The Common event is implemented.                        | 0b1   |



| Bits | Name | Description                                                                                                      | Reset |
|------|------|------------------------------------------------------------------------------------------------------------------|-------|
| [3]  | ID3  | ID3 corresponds to common event (0x23) STALL_FRONTEND<br><br><b>0b1</b><br>The Common event is implemented.      | 0b1   |
| [2]  | ID2  | ID2 corresponds to common event (0x22) BR_MIS_PRED_RETIRED<br><br><b>0b1</b><br>The Common event is implemented. | 0b1   |
| [1]  | ID1  | ID1 corresponds to common event (0x21) BR_RETIRED<br><br><b>0b1</b><br>The Common event is implemented.          | 0b1   |
| [0]  | ID0  | ID0 corresponds to common event (0x20) L2D_CACHE_ALLOCATE<br><br><b>0b1</b><br>The Common event is implemented.  | 0b1   |

## Access

MRS <Xt>, PMCEID1\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1001 | 0b1100 | 0b111 |

## Accessibility

MRS <Xt>, PMCEID1\_ELO

```

if PSTATE.EL == EL0 then
    if PMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HDFGRTR_EL2.PMCEIDn_EL0 == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMCEID1_EL0;
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.PMCEIDn_EL0 == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif MDCR_EL3.TPM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMCEID1_EL0;
        elsif PSTATE.EL == EL2 then
            if MDCR_EL3.TPM == '1' then
                if Halted() && EDSCR.SDD == '1' then

```

```

        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMCEID1_EL0;
elseif PSTATE.EL == EL3 then
    X[t, 64] = PMCEID1_EL0;

```

## A.8 AArch64 GIC system registers summary

The following summary table provides an overview of all GIC system registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-238: GIC system registers summary**

| Name                           | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                                |
|--------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------------------|
| <a href="#">ICC_PMR_EL1</a>    | 3   | 0   | C4  | C6  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Priority Mask Register                      |
| <a href="#">ICV_PMR_EL1</a>    | 3   | 0   | C4  | C6  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Priority Mask Register              |
| <a href="#">ICC_IAR0_EL1</a>   | 3   | 0   | C12 | C8  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Acknowledge Register 0                      |
| <a href="#">ICV_IAR0_EL1</a>   | 3   | 0   | C12 | C8  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Acknowledge Register 0              |
| <a href="#">ICC_EOIRO_EL1</a>  | 3   | 0   | C12 | C8  | 1   | See individual bit resets. | 64-bit | Interrupt Controller End Of Interrupt Register 0                           |
| <a href="#">ICV_EOIRO_EL1</a>  | 3   | 0   | C12 | C8  | 1   | See individual bit resets. | 64-bit | Interrupt Controller Virtual End Of Interrupt Register 0                   |
| <a href="#">ICC_HPPIRO_EL1</a> | 3   | 0   | C12 | C8  | 2   | See individual bit resets. | 64-bit | Interrupt Controller Highest Priority Pending Interrupt Register 0         |
| <a href="#">ICV_HPPIRO_EL1</a> | 3   | 0   | C12 | C8  | 2   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Highest Priority Pending Interrupt Register 0 |
| <a href="#">ICV_BPRO_EL1</a>   | 3   | 0   | C12 | C8  | 3   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Binary Point Register 0                       |
| <a href="#">ICC_AP0R0_EL1</a>  | 3   | 0   | C12 | C8  | 4   | See individual bit resets. | 64-bit | Interrupt Controller Active Priorities Group 0 Registers                   |
| <a href="#">ICV_AP0R0_EL1</a>  | 3   | 0   | C12 | C8  | 4   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Active Priorities Group 0 Registers           |
| <a href="#">ICC_AP1R0_EL1</a>  | 3   | 0   | C12 | C9  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Active Priorities Group 1 Registers                   |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                                |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------------------|
| ICV_AP1R0_EL1   | 3   | 0   | C12 | C9  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Active Priorities Group 1 Registers           |
| ICC_DIR_EL1     | 3   | 0   | C12 | C11 | 1   | See individual bit resets. | 64-bit | Interrupt Controller Deactivate Interrupt Register                         |
| ICV_DIR_EL1     | 3   | 0   | C12 | C11 | 1   | See individual bit resets. | 64-bit | Interrupt Controller Deactivate Virtual Interrupt Register                 |
| ICC_RPR_EL1     | 3   | 0   | C12 | C11 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Running Priority Register                             |
| ICV_RPR_EL1     | 3   | 0   | C12 | C11 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Running Priority Register                     |
| ICC_SGI1R_EL1   | 3   | 0   | C12 | C11 | 5   | See individual bit resets. | 64-bit | Interrupt Controller Software Generated Interrupt Group 1 Register         |
| ICC_ASIG1R_EL1  | 3   | 0   | C12 | C11 | 6   | See individual bit resets. | 64-bit | Interrupt Controller Alias Software Generated Interrupt Group 1 Register   |
| ICC_SGI0R_EL1   | 3   | 0   | C12 | C11 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Software Generated Interrupt Group 0 Register         |
| ICC_IAR1_EL1    | 3   | 0   | C12 | C12 | 0   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Acknowledge Register 1                      |
| ICV_IAR1_EL1    | 3   | 0   | C12 | C12 | 0   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Acknowledge Register 1              |
| ICC_EOIR1_EL1   | 3   | 0   | C12 | C12 | 1   | See individual bit resets. | 64-bit | Interrupt Controller End Of Interrupt Register 1                           |
| ICV_EOIR1_EL1   | 3   | 0   | C12 | C12 | 1   | See individual bit resets. | 64-bit | Interrupt Controller Virtual End Of Interrupt Register 1                   |
| ICC_HPPIR1_EL1  | 3   | 0   | C12 | C12 | 2   | See individual bit resets. | 64-bit | Interrupt Controller Highest Priority Pending Interrupt Register 1         |
| ICV_HPPIR1_EL1  | 3   | 0   | C12 | C12 | 2   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Highest Priority Pending Interrupt Register 1 |
| ICC_BPR1_EL1    | 3   | 0   | C12 | C12 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Binary Point Register 1                               |
| ICV_BPR1_EL1    | 3   | 0   | C12 | C12 | 3   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Binary Point Register 1                       |
| ICC_CTLR_EL1    | 3   | 0   | C12 | C12 | 4   | See individual bit resets. | 64-bit | Interrupt Controller Control Register (EL1)                                |
| ICV_CTLR_EL1    | 3   | 0   | C12 | C12 | 4   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Control Register                              |
| ICC_SRE_EL1     | 3   | 0   | C12 | C12 | 5   | See individual bit resets. | 64-bit | Interrupt Controller System Register Enable Register (EL1)                 |
| ICC_IGRPEN0_EL1 | 3   | 0   | C12 | C12 | 6   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Group 0 Enable Register                     |
| ICV_IGRPEN0_EL1 | 3   | 0   | C12 | C12 | 6   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Group 0 Enable Register             |
| ICC_IGRPEN1_EL1 | 3   | 0   | C12 | C12 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Group 1 Enable Register                     |
| ICV_IGRPEN1_EL1 | 3   | 0   | C12 | C12 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Interrupt Group 1 Enable Register             |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                  |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------------------|
| ICH_APOR0_EL2   | 3   | 4   | C12 | C8  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Hyp Active Priorities Group 0 Registers |
| ICH_AP1R0_EL2   | 3   | 4   | C12 | C9  | 0   | See individual bit resets. | 64-bit | Interrupt Controller Hyp Active Priorities Group 1 Registers |
| ICC_SRE_EL2     | 3   | 4   | C12 | C9  | 5   | See individual bit resets. | 64-bit | Interrupt Controller System Register Enable Register (EL2)   |
| ICH_HCR_EL2     | 3   | 4   | C12 | C11 | 0   | See individual bit resets. | 64-bit | Interrupt Controller Hyp Control Register                    |
| ICH_VTR_EL2     | 3   | 4   | C12 | C11 | 1   | See individual bit resets. | 64-bit | Interrupt Controller VGIC Type Register                      |
| ICH_MISR_EL2    | 3   | 4   | C12 | C11 | 2   | See individual bit resets. | 64-bit | Interrupt Controller Maintenance Interrupt State Register    |
| ICH_EISR_EL2    | 3   | 4   | C12 | C11 | 3   | See individual bit resets. | 64-bit | Interrupt Controller End of Interrupt Status Register        |
| ICH_ELSR_EL2    | 3   | 4   | C12 | C11 | 5   | See individual bit resets. | 64-bit | Interrupt Controller Empty List Register Status Register     |
| ICH_VMCR_EL2    | 3   | 4   | C12 | C11 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Virtual Machine Control Register        |
| ICH_LR0_EL2     | 3   | 4   | C12 | C12 | 0   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                          |
| ICH_LR1_EL2     | 3   | 4   | C12 | C12 | 1   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                          |
| ICH_LR2_EL2     | 3   | 4   | C12 | C12 | 2   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                          |
| ICH_LR3_EL2     | 3   | 4   | C12 | C12 | 3   | See individual bit resets. | 64-bit | Interrupt Controller List Registers                          |
| ICC_CTLR_EL3    | 3   | 6   | C12 | C12 | 4   | See individual bit resets. | 64-bit | Interrupt Controller Control Register (EL3)                  |
| ICC_SRE_EL3     | 3   | 6   | C12 | C12 | 5   | See individual bit resets. | 64-bit | Interrupt Controller System Register Enable Register (EL3)   |
| ICC_IGRPEN1_EL3 | 3   | 6   | C12 | C12 | 7   | See individual bit resets. | 64-bit | Interrupt Controller Interrupt Group 1 Enable Register (EL3) |

## A.8.1 ICC\_APOR0\_EL1, Interrupt Controller Active Priorities Group 0 Registers

Provides information about Group 0 active priorities.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

Functional group

GIC system registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|---|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |   |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0 |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-86: AArch64\_icc\_ap0r0\_el1 bit assignments

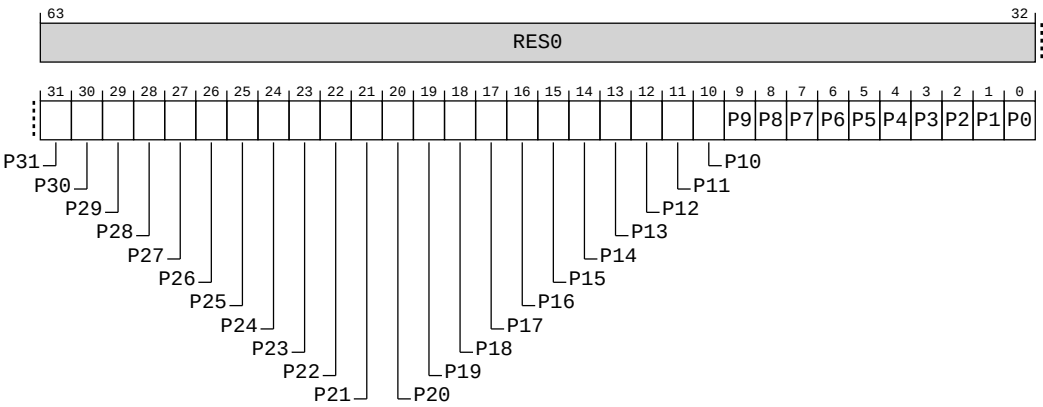


Table A-239: ICC\_AP0R0\_EL1 bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RES0  |
| [31]    | P31  | Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.<br><br>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3]. | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [30] | P30  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [29] | P29  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [28] | P28  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [27] | P27  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [26] | P26  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [25] | P25  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [24] | P24  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [23] | P23  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [22] | P22  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [21] | P21  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [20] | P20  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [19] | P19  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [18] | P18  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [17] | P17  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [16] | P16  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |



| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15] | P15  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [14] | P14  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [13] | P13  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [12] | P12  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [11] | P11  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10] | P10  | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [9]  | P9   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [8]  | P8   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [7]  | P7   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [6]  | P6   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [5]  | P5   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [4]  | P4   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [3]  | P3   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [2]  | P2   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [1]  | P1   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | PO   | <p>Provides the access to the active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

The contents of these registers are **IMPLEMENTATION DEFINED** with the one architectural requirement that the value 0x00000000 is consistent with no interrupts being active.

### Access

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 0 active priorities) might result in **UNPREDICTABLE** behavior of the interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICC\_AP0R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICC\_AP0R2\_EL1 and ICC\_AP0R3\_EL1 are implemented only in implementations that support 7 or more bits of priority. Unimplemented registers are **UNDEFINED**.



The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits.

Writing to the active priority registers in any order other than the following order will result in **UNPREDICTABLE** behavior:

- ICC\_AP0R<n>\_EL1.
- Secure AArch64-ICC\_AP1R<n>\_EL1.
- Non-secure AArch64-ICC\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP0R0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1000 | 0b100 |

MSR ICC\_AP0R0\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1000 | 0b100 |

## Accessibility

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 0 active priorities) might result in UNPREDICTABLE behavior of the interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICC\_AP0R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICC\_AP0R2\_EL1 and ICC\_AP0R3\_EL1 are implemented only in implementations that support 7 or more bits of priority. Unimplemented registers are UNDEFINED.



Note

The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits.

Writing to the active priority registers in any order other than the following order will result in UNPREDICTABLE behavior:

- ICC\_AP0R<n>\_EL1.
- Secure AArch64-ICC\_AP1R<n>\_EL1.
- Non-secure AArch64-ICC\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP0R0\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        X[t, 64] = ICV_AP0R_EL1[0];
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ICC_AP0R_EL1[0];
    elseif PSTATE.EL == EL2 then
        if ICC_SRE_EL2.SRE == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif SCR_EL3.FIQ == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ICC_AP0R_EL1[0];
    elseif PSTATE.EL == EL3 then
        if ICC_SRE_EL3.SRE == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ICC_AP0R_EL1[0];
```

## MSR ICC\_AP0R0\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        ICV_AP0R_EL1[0] = X[t, 64];
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICC_AP0R_EL1[0] = X[t, 64];
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICC_AP0R_EL1[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICC_AP0R_EL1[0] = X[t, 64];

```

## A.8.2 ICV\_AP0R0\_EL1, Interrupt Controller Virtual Active Priorities Group 0 Registers

Provides information about virtual Group 0 active priorities.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

GIC system registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-87: AArch64\_icv\_ap0r0\_el1 bit assignments**

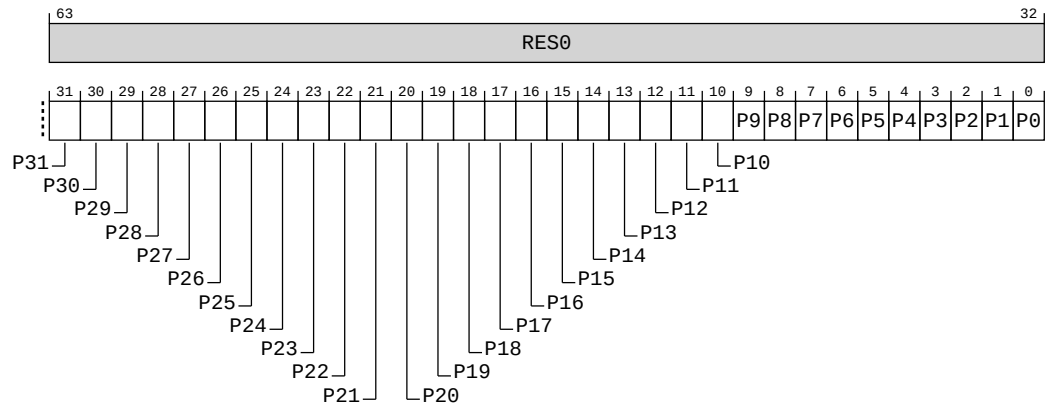


**Table A-242: ICV\_AP0R0\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RES0  |
| [31]    | P31  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.<br><br>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3]. | x     |
| [30]    | P30  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.<br><br>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3]. | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [29] | P29  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [28] | P28  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [27] | P27  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [26] | P26  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [25] | P25  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |



| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [24] | P24  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [23] | P23  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [22] | P22  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [21] | P21  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [20] | P20  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [19] | P19  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [18] | P18  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [17] | P17  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [16] | P16  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [15] | P15  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [14] | P14  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [13] | P13  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [12] | P12  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [11] | P11  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [10] | P10  | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [9]  | P9   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [8]  | P8   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [7]  | P7   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [6]  | P6   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [5]  | P5   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [4]  | P4   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [3]  | P3   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [2]  | P2   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [1]  | P1   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [0]  | P0   | <p>Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 0 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

The contents of these registers are **IMPLEMENTATION DEFINED** with the one architectural requirement that the value 0x00000000 is consistent with no interrupts being active.

### Access

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 0 active priorities) might result in **UNPREDICTABLE** behavior of the virtual interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICV\_AP0R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICV\_AP0R2\_EL1 and ICV\_AP0R3\_EL1 are implemented only in implementations that support 7 bits of priority. Unimplemented registers are **UNDEFINED**.

Writing to the active priority registers in any order other than the following order might result in **UNPREDICTABLE** behavior of the interrupt prioritization system:

- ICV\_AP0R<n>\_EL1.
- AArch64-ICV\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP0R0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1000 | 0b100 |

MSR ICC\_AP0R0\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1000 | 0b100 |

### Accessibility

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 0 active priorities) might result in **UNPREDICTABLE** behavior of the virtual interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICV\_AP0R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICV\_AP0R2\_EL1 and ICV\_AP0R3\_EL1 are implemented only in implementations that support 7 bits of priority. Unimplemented registers are **UNDEFINED**.

Writing to the active priority registers in any order other than the following order might result in **UNPREDICTABLE** behavior of the interrupt prioritization system:

- ICV\_AP0R<n>\_EL1.
- AArch64-ICV\_AP1R<n>\_EL1.

## MRS &lt;Xt&gt;, ICC\_APOR0\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        X[t, 64] = ICV_APOR_EL1[0];
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_APOR_EL1[0];
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_APOR_EL1[0];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_APOR_EL1[0];

```

## MSR ICC\_APOR0\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        ICV_APOR_EL1[0] = X[t, 64];
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICC_APOR_EL1[0] = X[t, 64];
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.FIQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICC_APOR_EL1[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else

```

```
ICC_AP0R_EL1[0] = X[t, 64];
```

### A.8.3 ICC\_AP1R0\_EL1, Interrupt Controller Active Priorities Group 1 Registers

Provides information about Group 1 active priorities.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

GIC system registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-88: AArch64\_icc\_ap1r0\_el1 bit assignments

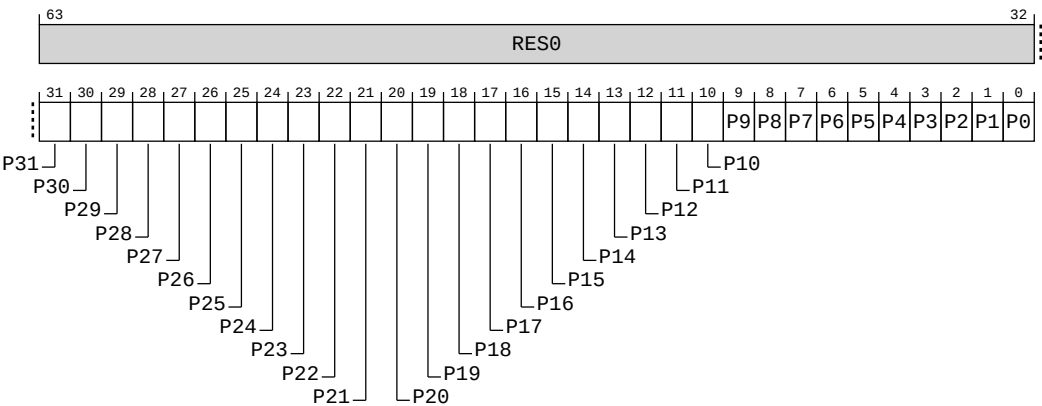




**Table A-245: ICC\_AP1R0\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|---------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0  |
| [31]    | P31  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [30]    | P30  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [29]    | P29  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [28] | P28  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [27] | P27  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [26] | P26  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [25] | P25  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [24] | P24  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [23] | P23  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [22] | P22  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [21] | P21  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [20] | P20  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [19] | P19  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [18] | P18  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [17] | P17  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [16] | P16  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [15] | P15  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [14] | P14  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [13] | P13  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [12] | P12  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [11] | P11  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10] | P10  | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [9]  | P9   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [8]  | P8   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |



| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7]  | P7   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [6]  | P6   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [5]  | P5   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [4]  | P4   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [3]  | P3   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [2]  | P2   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [1]  | P1   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |
| [0]  | P0   | <p>Group 1 interrupt active priorities. When AArch64-SCR_EL3.NS == '1', accesses the priorities for Non-secure Group 1 interrupts, and when AArch64-SCR_EL3.NS == '0' accesses the priorities for Secure Group 1 interrupts. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> <p>When accessed from non-secure EL2 or EL1, only the 16 lowest-priority interrupts are visible in bits [15:0] of this register.</p> | x     |

The contents of these registers are **IMPLEMENTATION DEFINED** with the one architectural requirement that the value 0x00000000 is consistent with no interrupts being active.

## Access

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 1 active priorities) might result in **UNPREDICTABLE** behavior of the interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICC\_AP1R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICC\_AP1R2\_EL1 and ICC\_AP1R3\_EL1 are implemented only in implementations that support 7 or more bits of priority. Unimplemented registers are **UNDEFINED**.



Note

The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits.

Writing to the active priority registers in any order other than the following order will result in **UNPREDICTABLE** behavior:

- AArch64-ICC\_AP0R<n>\_EL1.
- Secure ICC\_AP1R<n>\_EL1.
- Non-secure ICC\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP1R0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1001 | 0b000 |

MSR ICC\_AP1R0\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1001 | 0b000 |

## Accessibility

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 1 active priorities) might result in UNPREDICTABLE behavior of the interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICC\_AP1R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICC\_AP1R2\_EL1 and ICC\_AP1R3\_EL1 are implemented only in implementations that support 7 or more bits of priority. Unimplemented registers are UNDEFINED.



Note

The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits.

Writing to the active priority registers in any order other than the following order will result in UNPREDICTABLE behavior:

- AArch64-ICC\_AP0R<n>\_EL1.
- Secure ICC\_AP1R<n>\_EL1.
- Non-secure ICC\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP1R0\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL1 == '1' then
```

```

    AArch64.SystemAccessTrap(EL2, 0x18);
elseif EL2Enabled() && HCR_EL2.IMO == '1' then
    X[t, 64] = ICV_AP1R_EL1[0];
elseif SCR_EL3.IRQ == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
else
    if SCR_EL3.NS == '0' then
        X[t, 64] = ICC_AP1R_EL1_S[0];
    else
        X[t, 64] = ICC_AP1R_EL1_NS[0];
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
elseif SCR_EL3.IRQ == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
else
    if SCR_EL3.NS == '0' then
        X[t, 64] = ICC_AP1R_EL1_S[0];
    else
        X[t, 64] = ICC_AP1R_EL1_NS[0];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_AP1R_EL1_S[0];
        else
            X[t, 64] = ICC_AP1R_EL1_NS[0];

```

## MSR ICC\_AP1R0\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        ICV_AP1R_EL1[0] = X[t, 64];
    elseif SCR_EL3.IRQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_AP1R_EL1_S[0] = X[t, 64];
        else
            ICC_AP1R_EL1_NS[0] = X[t, 64];
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.IRQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_AP1R_EL1_S[0] = X[t, 64];
        else

```

```
        ICC_AP1R_EL1_NS[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_AP1R_EL1_S[0] = X[t, 64];
        else
            ICC_AP1R_EL1_NS[0] = X[t, 64];
```

## A.8.4 ICV\_AP1R0\_EL1, Interrupt Controller Virtual Active Priorities Group 1 Registers

Provides information about virtual Group 1 active priorities.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

GIC system registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-89: AArch64\_icv\_ap1r0\_el1 bit assignments**

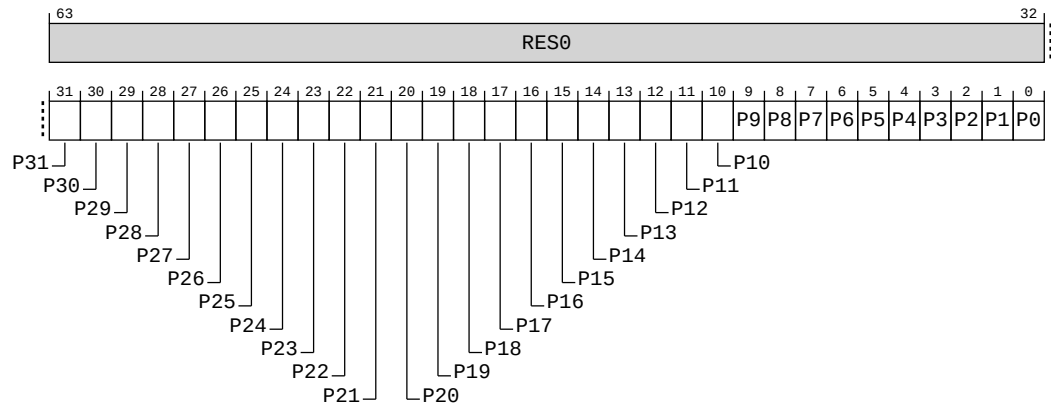


**Table A-248: ICV\_AP1R0\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RES0  |
| [31]    | P31  | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.<br><br>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3]. | x     |
| [30]    | P30  | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.<br><br>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3]. | x     |
| [29]    | P29  | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.<br><br>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3]. | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [28] | P28  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [27] | P27  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [26] | P26  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [25] | P25  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [24] | P24  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |



| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [23] | P23  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [22] | P22  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [21] | P21  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [20] | P20  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [19] | P19  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [18] | P18  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [17] | P17  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [16] | P16  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [15] | P15  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [14] | P14  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [13] | P13  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [12] | P12  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [11] | P11  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [10] | P10  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [9]  | P9   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [8]  | P8   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [7]  | P7   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [6]  | P6   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [5]  | P5   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [4]  | P4   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [3]  | P3   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [2]  | P2   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [1]  | P1   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |
| [0]  | P0   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> <p>There are 32 preemption levels, and the active state of these preemption levels are held in the bits corresponding to Priority[7:3].</p> | x     |

The contents of these registers are **IMPLEMENTATION DEFINED** with the one architectural requirement that the value 0x00000000 is consistent with no interrupts being active.

## Access

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 1 active priorities) might result in **UNPREDICTABLE** behavior of the virtual interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICV\_AP1R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICV\_AP1R2\_EL1 and ICV\_AP1R3\_EL1 are implemented only in implementations that support 7 bits of priority. Unimplemented registers are **UNDEFINED**.

Writing to the active priority registers in any order other than the following order might result in **UNPREDICTABLE** behavior of the interrupt prioritization system:

- AArch64-ICV\_APOR<n>\_EL1.
- ICV\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP1R0\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1001 | 0b000 |

MSR ICC\_AP1R0\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1001 | 0b000 |

## Accessibility

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 1 active priorities) might result in **UNPREDICTABLE** behavior of the virtual interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICV\_AP1R1\_EL1 is implemented only in implementations that support 6 or more bits of priority. ICV\_AP1R2\_EL1 and ICV\_AP1R3\_EL1 are implemented only in implementations that support 7 bits of priority. Unimplemented registers are **UNDEFINED**.

Writing to the active priority registers in any order other than the following order might result in **UNPREDICTABLE** behavior of the interrupt prioritization system:

- AArch64-ICV\_APOR<n>\_EL1.
- ICV\_AP1R<n>\_EL1.

MRS <Xt>, ICC\_AP1R0\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TALL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        X[t, 64] = ICV_AP1R_EL1[0];
    elseif SCR_EL3.IRQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else

```

```

        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_AP1R_EL1_S[0];
        else
            X[t, 64] = ICC_AP1R_EL1_NS[0];
    elseif PSTATE.EL == EL2 then
        if ICC_SRE_EL2.SRE == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif SCR_EL3.IRQ == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                X[t, 64] = ICC_AP1R_EL1_S[0];
            else
                X[t, 64] = ICC_AP1R_EL1_NS[0];
    elseif PSTATE.EL == EL3 then
        if ICC_SRE_EL3.SRE == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                X[t, 64] = ICC_AP1R_EL1_S[0];
            else
                X[t, 64] = ICC_AP1R_EL1_NS[0];

```

## MSR ICC\_AP1RO\_EL1, &lt;Xt&gt;

```

    if PSTATE.EL == EL0 then
        UNDEFINED;
    elseif PSTATE.EL == EL1 then
        if ICC_SRE_EL1.SRE == '0' then
            AArch64.SystemAccessTrap(EL1, 0x18);
        elseif EL2Enabled() && ICH_HCR_EL2.TALL1 == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif EL2Enabled() && HCR_EL2.IMO == '1' then
            ICV_AP1R_EL1[0] = X[t, 64];
        elseif SCR_EL3.IRQ == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                ICC_AP1R_EL1_S[0] = X[t, 64];
            else
                ICC_AP1R_EL1_NS[0] = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if ICC_SRE_EL2.SRE == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif SCR_EL3.IRQ == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                ICC_AP1R_EL1_S[0] = X[t, 64];
            else
                ICC_AP1R_EL1_NS[0] = X[t, 64];
    elseif PSTATE.EL == EL3 then
        if ICC_SRE_EL3.SRE == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                ICC_AP1R_EL1_S[0] = X[t, 64];

```

```
else
    ICC_AP1R_EL1_NS[0] = X[t, 64];
```

A.8.5 ICC\_CTLR\_EL1, Interrupt Controller Control Register (EL1)

Controls aspects of the behavior of the GIC CPU interface and provides information about the features implemented.

Configurations

This register is available in all configurations.

Attributes

Width

64

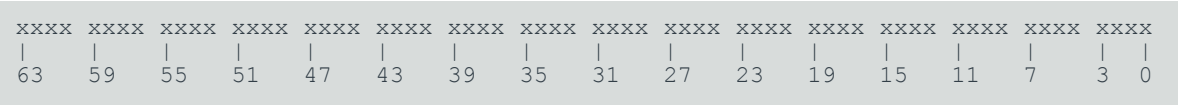
Functional group

GIC system registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Note

Bit descriptions

Figure A-90: AArch64\_icc\_ctlr\_el1 bit assignments

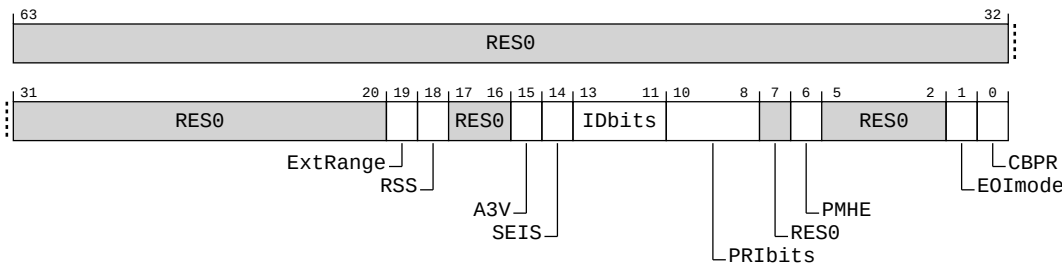


Table A-251: ICC\_CTLR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:20] | RES0 | Reserved    | RES0  |



| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [19]    | ExtRange | Extended INTID range (read-only).<br><br><b>0b1</b><br>CPU interface supports INTIDs in the range 1024..8191<br><ul style="list-style-type: none"> <li>All INTIDs in the range 1024..8191 are treated as requiring deactivation.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | x     |
| [18]    | RSS      | Range Selector Support. Possible values are:<br><br><b>0b0</b><br>Targeted SGLs with affinity level 0 values of 0 - 15 are supported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | x     |
| [17:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0  |
| [15]    | A3V      | Affinity 3 Valid. Read-only and writes are ignored. Possible values are:<br><br><b>0b1</b><br>The CPU interface logic supports nonzero values of Affinity 3 in SGI generation System registers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | x     |
| [14]    | SEIS     | SEI Support. Read-only and writes are ignored. Indicates whether the CPU interface supports local generation of SEIs:<br><br><b>0b0</b><br>The CPU interface logic does not support local generation of SEIs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | x     |
| [13:11] | IDbits   | Identifier bits. Read-only and writes are ignored. The number of physical interrupt identifier bits supported:<br><br><b>0b000</b><br>16 bits.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | xxx   |
| [10:8]  | PRIbits  | Priority bits. Read-only and writes are ignored. The number of priority bits implemented, minus one.<br><br>An implementation that supports two Security states must implement at least 32 levels of physical priority (5 priority bits).<br><br>An implementation that supports only a single Security state must implement at least 16 levels of physical priority (4 priority bits).<br><br><b>Note:</b><br>This field always returns the number of priority bits implemented, regardless of the Security state of the access or the value of ext-GICD_CTLR.DS.<br><br>For physical accesses, this field determines the minimum value of AArch64-ICC_BPR0_EL1.<br><br>If EL3 is implemented, physical accesses return the value from AArch64-ICC_CTLR_EL3.PRIbits.<br><br><b>0b100</b><br>5 bits of priority are implemented | xxx   |
| [7]     | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0  |
| [6]     | PMHE     | Priority Mask Hint Enable. Controls whether the priority mask register is used as a hint for interrupt distribution:<br><br><b>0b0</b><br>Disables use of AArch64-ICC_PMR_EL1 as a hint for interrupt distribution.<br><br><b>0b1</b><br>Enables use of AArch64-ICC_PMR_EL1 as a hint for interrupt distribution.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | x     |
| [5:2]   | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0  |

| Bits | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [1]  | EOImode | EOI mode for the current Security state. Controls whether a write to an End of Interrupt register also deactivates the interrupt:<br><br><b>0b0</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide both priority drop and interrupt deactivation functionality. Accesses to AArch64-ICC_DIR_EL1 are <b>UNPREDICTABLE</b> .<br><br><b>0b1</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide priority drop functionality only. AArch64-ICC_DIR_EL1 provides interrupt deactivation functionality. | x     |
| [0]  | CBPR    | Common Binary Point Register. Controls whether the same register is used for interrupt preemption of both Group 0 and Group 1 interrupts:<br><br><b>0b0</b><br>AArch64-ICC_BPRO_EL1 determines the preemption group for Group 0 interrupts only.<br><br>AArch64-ICC_BPR1_EL1 determines the preemption group for Group 1 interrupts.<br><br><b>0b1</b><br>AArch64-ICC_BPRO_EL1 determines the preemption group for both Group 0 and Group 1 interrupts.                                                                  | x     |

## Access

MRS <Xt>, ICC\_CTLR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1100 | 0b100 |

MSR ICC\_CTLR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1100 | 0b100 |

## Accessibility

MRS <Xt>, ICC\_CTLR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        X[t, 64] = ICV_CTLR_EL1;
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        X[t, 64] = ICV_CTLR_EL1;
    elseif SCR_EL3.<IRQ,FIQ> == '11' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_CTLR_EL1_S;
        else
            X[t, 64] = ICC_CTLR_EL1_NS;

```

```

elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.<IRQ,FIQ> == '11' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                X[t, 64] = ICC_CTLR_EL1_S;
            else
                X[t, 64] = ICC_CTLR_EL1_NS;
    elseif PSTATE.EL == EL3 then
        if ICC_SRE_EL3.SRE == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                X[t, 64] = ICC_CTLR_EL1_S;
            else
                X[t, 64] = ICC_CTLR_EL1_NS;

```

## MSR ICC\_CTLR\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        ICV_CTLR_EL1 = X[t, 64];
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        ICV_CTLR_EL1 = X[t, 64];
    elseif SCR_EL3.<IRQ,FIQ> == '11' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_CTLR_EL1_S = X[t, 64];
        else
            ICC_CTLR_EL1_NS = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if ICC_SRE_EL2.SRE == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif SCR_EL3.<IRQ,FIQ> == '11' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                ICC_CTLR_EL1_S = X[t, 64];
            else
                ICC_CTLR_EL1_NS = X[t, 64];
    elseif PSTATE.EL == EL3 then
        if ICC_SRE_EL3.SRE == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then
                ICC_CTLR_EL1_S = X[t, 64];
            else
                ICC_CTLR_EL1_NS = X[t, 64];

```

A.8.6 ICV\_CTLR\_EL1, Interrupt Controller Virtual Control Register

Controls aspects of the behavior of the GIC virtual CPU interface and provides information about the features implemented.

Configurations

This register is available in all configurations.

Attributes

Width

64

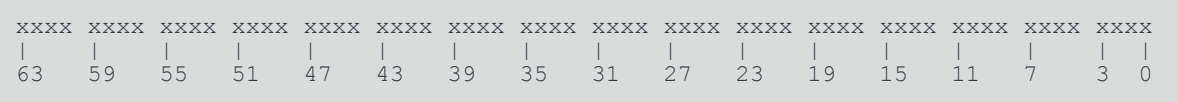
Functional group

GIC system registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-91: AArch64\_icv\_ctlr\_el1 bit assignments

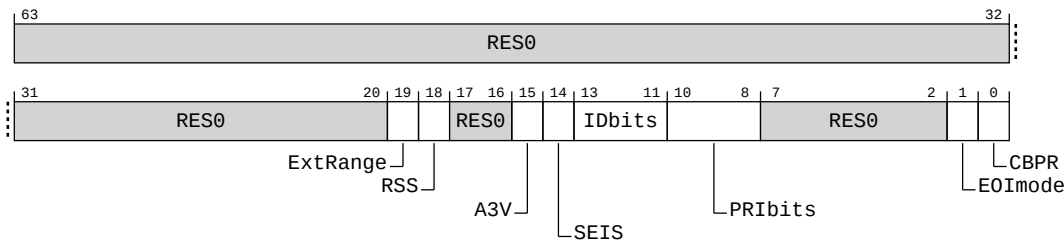


Table A-254: ICV\_CTLR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:20] | RES0 | Reserved    | RES0  |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset |
|---------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [19]    | ExtRange | Extended INTID range (read-only).<br><br><b>0b1</b><br>CPU interface supports INTIDs in the range 1024..8191<br><ul style="list-style-type: none"> <li>All INTIDs in the range 1024..8191 are treated as requiring deactivation.</li> </ul>                                                                                                                                                                                                                                                                    | x     |
| [18]    | RSS      | Range Selector Support. Possible values are:<br><br><b>0b0</b><br>Targeted SGLs with affinity level 0 values of 0 - 15 are supported.                                                                                                                                                                                                                                                                                                                                                                          | x     |
| [17:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RES0  |
| [15]    | A3V      | Affinity 3 Valid. Read-only and writes are ignored. Possible values are:<br><br><b>0b1</b><br>The virtual CPU interface logic supports nonzero values of Affinity 3 in SGI generation System registers.                                                                                                                                                                                                                                                                                                        | x     |
| [14]    | SEIS     | SEI Support. Read-only and writes are ignored. Indicates whether the virtual CPU interface supports local generation of SEIs:<br><br><b>0b0</b><br>The virtual CPU interface logic does not support local generation of SEIs.                                                                                                                                                                                                                                                                                  | x     |
| [13:11] | IDbits   | Identifier bits. Read-only and writes are ignored. The number of virtual interrupt identifier bits supported:<br><br><b>0b000</b><br>16 bits.                                                                                                                                                                                                                                                                                                                                                                  | xxx   |
| [10:8]  | PRIbits  | Priority bits. Read-only and writes are ignored. The number of priority bits implemented, minus one.<br><br>An implementation must implement at least 32 levels of physical priority (5 priority bits).<br><br><b>Note:</b><br>This field always returns the number of priority bits implemented.<br><br>The division between group priority and subpriority is defined in the binary point registers AArch64-ICV_BPR0_EL1 and AArch64-ICV_BPR1_EL1.<br><br><b>0b100</b><br>5 bits of priority are implemented | xxx   |
| [7:2]   | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RES0  |
| [1]     | EOImode  | Virtual EOI mode. Controls whether a write to an End of Interrupt register also deactivates the virtual interrupt:<br><br><b>0b0</b><br>AArch64-ICV_EOIR0_EL1 and AArch64-ICV_EOIR1_EL1 provide both priority drop and interrupt deactivation functionality. Accesses to AArch64-ICV_DIR_EL1 are <b>UNPREDICTABLE</b> .<br><br><b>0b1</b><br>AArch64-ICV_EOIR0_EL1 and AArch64-ICV_EOIR1_EL1 provide priority drop functionality only. AArch64-ICV_DIR_EL1 provides interrupt deactivation functionality.      | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | CBPR | <p>Common Binary Point Register. Controls whether the same register is used for interrupt preemption of both virtual Group 0 and virtual Group 1 interrupts:</p> <p><b>0b0</b></p> <p>AArch64-ICV_BPR1_EL1 determines the preemption group for virtual Group 1 interrupts.</p> <p><b>0b1</b></p> <p>Non-secure reads of AArch64-ICV_BPR1_EL1 return AArch64-ICV_BPR0_EL1 plus one, saturated to 0b111. Non-secure writes to AArch64-ICV_BPR1_EL1 are ignored.</p> <p>Secure reads of AArch64-ICV_BPR1_EL1 return AArch64-ICV_BPR0_EL1. Secure writes of AArch64-ICV_BPR1_EL1 modify AArch64-ICV_BPR0_EL1.</p> | x     |

## Access

MRS <Xt>, ICC\_CTLR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1100 | 0b100 |

MSR ICC\_CTLR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1100 | 0b1100 | 0b100 |

## Accessibility

MRS <Xt>, ICC\_CTLR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        X[t, 64] = ICV_CTLR_EL1;
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        X[t, 64] = ICV_CTLR_EL1;
    elseif SCR_EL3.<IRQ,FIQ> == '11' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_CTLR_EL1_S;
        else
            X[t, 64] = ICC_CTLR_EL1_NS;
    elseif PSTATE.EL == EL2 then
        if ICC_SRE_EL2.SRE == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif SCR_EL3.<IRQ,FIQ> == '11' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            if SCR_EL3.NS == '0' then

```

```

        X[t, 64] = ICC_CTLR_EL1_S;
    else
        X[t, 64] = ICC_CTLR_EL1_NS;
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_CTLR_EL1_S;
        else
            X[t, 64] = ICC_CTLR_EL1_NS;

```

## MSR ICC\_CTLR\_EL1, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        ICV_CTLR_EL1 = X[t, 64];
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        ICV_CTLR_EL1 = X[t, 64];
    elseif SCR_EL3.<IRQ,FIQ> == '11' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_CTLR_EL1_S = X[t, 64];
        else
            ICC_CTLR_EL1_NS = X[t, 64];
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.<IRQ,FIQ> == '11' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_CTLR_EL1_S = X[t, 64];
        else
            ICC_CTLR_EL1_NS = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            ICC_CTLR_EL1_S = X[t, 64];
        else
            ICC_CTLR_EL1_NS = X[t, 64];

```

## A.8.7 ICH\_AP0R0\_EL2, Interrupt Controller Hyp Active Priorities Group 0 Registers

Provides information about Group 0 virtual active priorities for EL2.

### Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

### Attributes

#### Width

64

#### Functional group

GIC system registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Note

Where the reset reads xxxx, see individual bits.

### Bit descriptions

Figure A-92: AArch64\_ich\_ap0r0\_el2 bit assignments

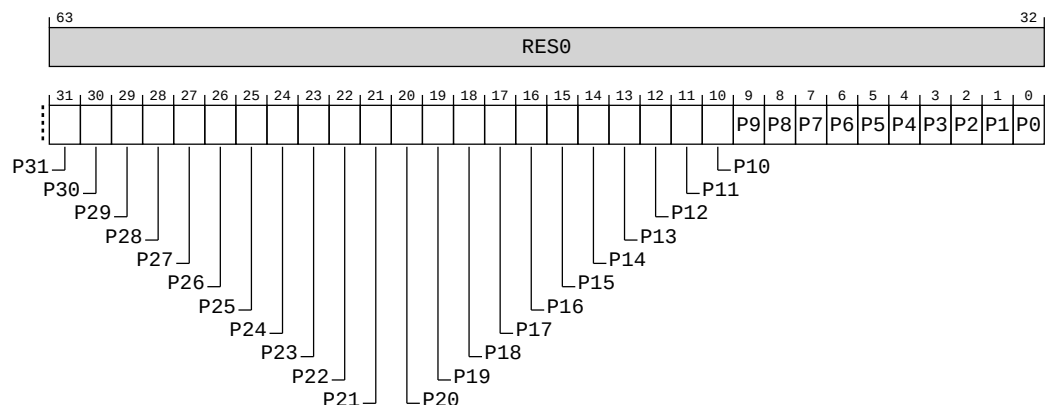




**Table A-257: ICH\_AP0R0\_EL2 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                           | RES0  |
| [31]    | P31  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [30]    | P30  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [29]    | P29  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [28]    | P28  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [27]    | P27  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [26]    | P26  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [25]    | P25  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [24] | P24  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [23] | P23  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [22] | P22  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [21] | P21  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [20] | P20  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [19] | P19  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [18] | P18  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [17] | P17  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [16] | P16  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [15] | P15  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [14] | P14  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [13] | P13  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [12] | P12  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [11] | P11  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10] | P10  | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [9]  | P9   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [8]  | P8   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [7]  | P7   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [6]  | P6   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [5]  | P5   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [4]  | P4   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [3]  | P3   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [2]  | P2   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [1]  | P1   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [0]  | P0   | Provides the access to the virtual active priorities for Group 0 interrupts. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 0 interrupt active with this priority level, or all active Group 0 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 0 interrupt active with this priority level which has not undergone priority drop. | 0b0   |

Software must ensure that ICH\_APOR<n>\_EL2 is 0 for legacy VMs otherwise behavior is **UNPREDICTABLE**. For more information about support for legacy VMs, see 'Support for legacy operation of VMs' in ARM® Generic Interrupt Controller Architecture Specification, GIC architecture version 3.0 and version 4.0 (ARM IHI 0069).

The active priorities for Group 0 and Group 1 interrupts for legacy VMs are held in AArch64-ICH\_AP1R<n>\_EL2 and reads and writes to GICV\_APR access AArch64-ICH\_AP1R<n>\_EL2. This means that ICH\_APOR<n>\_EL2 is inaccessible to legacy VMs.

### Access

ICH\_APOR1\_EL2 is implemented only in implementations that support 6 or more bits of preemption. ICH\_APOR2\_EL2 and ICH\_APOR3\_EL2 are implemented only in implementations that support 7 bits of preemption. Unimplemented registers are **UNDEFINED**.



Note

The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits

Writing to these registers with any value other than the last read value of the register (or 0x00000000 for a newly set up virtual machine) can result in **UNPREDICTABLE** behavior of the virtual interrupt prioritization system allowing either:

- Virtual interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution at EL1 or EL0.

Writing to the active priority registers in any order other than the following order will result in **UNPREDICTABLE** behavior:

- ICH\_AP0R<n>\_EL2.
- AArch64-ICH\_AP1R<n>\_EL2.

Having the bit corresponding to a priority set in both ICH\_AP0R<n>\_EL2 and AArch64-ICH\_AP1R<n>\_EL2 can result in **UNPREDICTABLE** behavior of the interrupt prioritization system for virtual interrupts.

MRS <Xt>, ICH\_AP0R0\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1000 | 0b000 |

MSR ICH\_AP0R0\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1000 | 0b000 |

## Accessibility

ICH\_AP0R1\_EL2 is implemented only in implementations that support 6 or more bits of preemption. ICH\_AP0R2\_EL2 and ICH\_AP0R3\_EL2 are implemented only in implementations that support 7 bits of preemption. Unimplemented registers are UNDEFINED.



Note

The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits

Writing to these registers with any value other than the last read value of the register (or 0x00000000 for a newly set up virtual machine) can result in UNPREDICTABLE behavior of the virtual interrupt prioritization system allowing either:

- Virtual interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution at EL1 or EL0.

Writing to the active priority registers in any order other than the following order will result in UNPREDICTABLE behavior:

- ICH\_AP0R<n>\_EL2.

- AArch64-ICH\_AP1R<n>\_EL2.

Having the bit corresponding to a priority set in both ICH\_AP0R<n>\_EL2 and AArch64-ICH\_AP1R<n>\_EL2 can result in UNPREDICTABLE behavior of the interrupt prioritization system for virtual interrupts.

MRS <Xt>, ICH\_AP0R0\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_AP0R_EL2[0];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_AP0R_EL2[0];
```

MSR ICH\_AP0R0\_EL2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_AP0R_EL2[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_AP0R_EL2[0] = X[t, 64];
```

## A.8.8 ICH\_AP1R0\_EL2, Interrupt Controller Hyp Active Priorities Group 1 Registers

Provides information about Group 1 virtual active priorities for EL2.

### Configurations

If EL2 is not implemented, this register is RES0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

### Attributes

#### Width

64

Functional group

GIC system registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-93: AArch64\_ich\_ap1r0\_el2 bit assignments

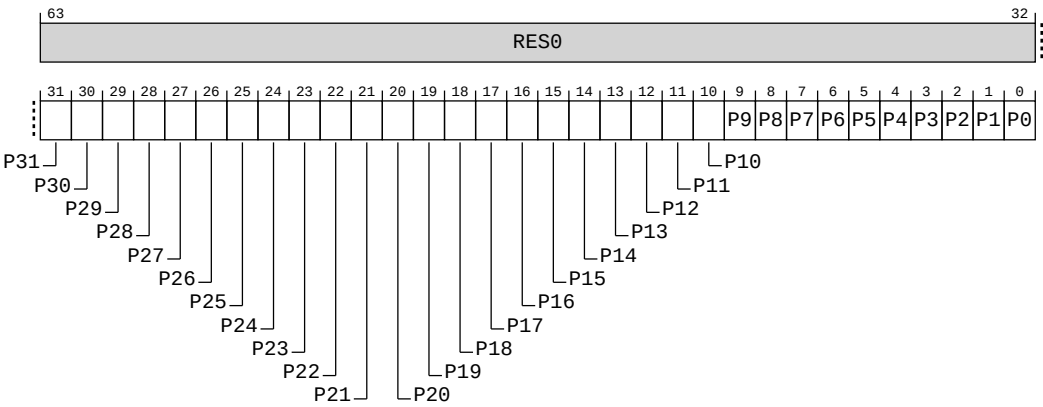


Table A-260: ICH\_AP1R0\_EL2 bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                | Reset |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                   | RES0  |
| [31]    | P31  | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop. | 0b0   |



| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [30] | P30  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [29] | P29  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [28] | P28  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [27] | P27  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [26] | P26  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [25] | P25  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [24] | P24  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [23] | P23  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [22] | P22  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [21] | P21  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [20] | P20  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [19] | P19  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [18] | P18  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [17] | P17  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [16] | P16  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [15] | P15  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [14] | P14  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [13] | P13  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [12] | P12  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [11] | P11  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [10] | P10  | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [9]  | P9   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [8]  | P8   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [7]  | P7   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [6]  | P6   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [5]  | P5   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [4]  | P4   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |
| [3]  | P3   | <p>Group 1 interrupt active priorities. Possible values of each bit are:</p> <p><b>0b0</b></p> <p>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.</p> <p><b>0b1</b></p> <p>There is a Group 1 interrupt active with this priority level which has not undergone priority drop.</p> | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                | Reset |
|------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [2]  | P2   | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [1]  | P1   | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop. | 0b0   |
| [0]  | P0   | Group 1 interrupt active priorities. Possible values of each bit are:<br><br><b>0b0</b><br>There is no Group 1 interrupt active with this priority level, or all active Group 1 interrupts with this priority level have undergone priority-drop.<br><br><b>0b1</b><br>There is a Group 1 interrupt active with this priority level which has not undergone priority drop. | 0b0   |

This register is always used for legacy VMs, regardless of the group of the virtual interrupt. Reads and writes to ext-GICV\_APR<n> access AArch64-ICH\_AP1R<n>\_EL2. For more information about support for legacy VMs, see 'Support for legacy operation of VMs' in ARM® Generic Interrupt Controller Architecture Specification, GIC architecture version 3.0 and version 4.0 (ARM IHI 0069).

### Access

ICH\_AP1R1\_EL2 is implemented only in implementations that support 6 or more bits of preemption. ICH\_AP1R2\_EL2 and ICH\_AP1R3\_EL2 are implemented only in implementations that support 7 bits of preemption. Unimplemented registers are **UNDEFINED**.



Note

The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits

Writing to these registers with any value other than the last read value of the register (or 0x00000000 for a newly set up virtual machine) can result in **UNPREDICTABLE** behavior of the virtual interrupt prioritization system allowing either:

Writing to the active priority registers in any order other than the following order will result in **UNPREDICTABLE** behavior:

MRS <Xt>, ICH\_AP1R0\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1001 | 0b000 |

## MSR ICH\_AP1R0\_EL2, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1001 | 0b000 |

**Accessibility**

ICH\_AP1R1\_EL2 is implemented only in implementations that support 6 or more bits of preemption. ICH\_AP1R2\_EL2 and ICH\_AP1R3\_EL2 are implemented only in implementations that support 7 bits of preemption. Unimplemented registers are UNDEFINED.



The number of bits of preemption is indicated by AArch64-ICH\_VTR\_EL2.PREbits

Writing to these registers with any value other than the last read value of the register (or 0x00000000 for a newly set up virtual machine) can result in UNPREDICTABLE behavior of the virtual interrupt prioritization system allowing either:

Writing to the active priority registers in any order other than the following order will result in UNPREDICTABLE behavior:

MRS <Xt>, ICH\_AP1R0\_EL2

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_AP1R_EL2[0];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_AP1R_EL2[0];

```

## MSR ICH\_AP1R0\_EL2, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_AP1R_EL2[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_AP1R_EL2[0] = X[t, 64];

```

A.8.9 ICH\_VTR\_EL2, Interrupt Controller VGIC Type Register

Reports supported GIC virtualization features.

Configurations

If EL2 is not implemented, all bits in this register are RES0 from EL3, except for nV4, which is RES1 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

Functional group

GIC system registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-94: AArch64\_ich\_vtr\_el2 bit assignments

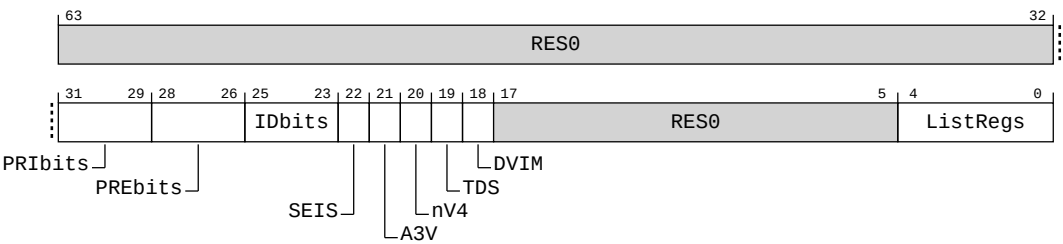


Table A-263: ICH\_VTR\_EL2 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:32] | RES0 | Reserved    | RES0  |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|---------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:29] | PRIbits  | <p>Priority bits. The number of virtual priority bits implemented, minus one.</p> <p>An implementation must implement at least 32 levels of virtual priority (5 priority bits).</p> <p>This field is an alias of AArch64-ICV_CTLR_EL1.PRIbits.</p> <p><b>0b100</b></p> <p>5 virtual priority bits are implemented</p>                                                                                                                                                                                                | xxx   |
| [28:26] | PREbits  | <p>The number of virtual preemption bits implemented, minus one.</p> <p>An implementation must implement at least 32 levels of virtual preemption priority (5 preemption bits).</p> <p>The value of this field must be less than or equal to the value of ICH_VTR_EL2.PRIbits.</p> <p>The maximum value of this field is 6, indicating 7 bits of preemption.</p> <p>This field determines the minimum value of AArch64-ICH_VMCR_EL2.VBPR0.</p> <p><b>0b100</b></p> <p>5 virtual pre-emption bits are implemented</p> | xxx   |
| [25:23] | IDbits   | <p>The number of virtual interrupt identifier bits supported:</p> <p><b>0b000</b></p> <p>16 bits.</p>                                                                                                                                                                                                                                                                                                                                                                                                                | xxx   |
| [22]    | SEIS     | <p>SEI Support. Indicates whether the virtual CPU interface supports generation of SEIs:</p> <p><b>0b0</b></p> <p>The virtual CPU interface logic does not support generation of SEIs.</p>                                                                                                                                                                                                                                                                                                                           | x     |
| [21]    | A3V      | <p>Affinity 3 Valid. Possible values are:</p> <p><b>0b1</b></p> <p>The virtual CPU interface logic supports nonzero values of Affinity 3 in SGI generation System registers.</p>                                                                                                                                                                                                                                                                                                                                     | x     |
| [20]    | nV4      | <p>Direct injection of virtual interrupts not supported. Possible values are:</p> <p><b>0b0</b></p> <p>The CPU interface logic supports direct injection of virtual interrupts.</p>                                                                                                                                                                                                                                                                                                                                  | x     |
| [19]    | TDS      | <p>Separate trapping of EL1 writes to AArch64-ICV_DIR_EL1 supported.</p> <p><b>0b1</b></p> <p>Implementation supports AArch64-ICH_HCR_EL2.TDIR.</p>                                                                                                                                                                                                                                                                                                                                                                  | x     |
| [18]    | DVIM     | <p>Masking of directly-injected virtual interrupts.</p> <p><b>0b0</b></p> <p>Masking of Directly-injected Virtual Interrupts not supported.</p>                                                                                                                                                                                                                                                                                                                                                                      | x     |
| [17:5]  | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RES0  |
| [4:0]   | ListRegs | <p>The number of implemented List registers, minus one. For example, a value of 0b01111 indicates that the maximum of 16 List registers are implemented.</p> <p><b>0b00011</b></p> <p>4 List registers</p>                                                                                                                                                                                                                                                                                                           | 5 {x} |

## Access

MRS <Xt>, ICH\_VTR\_EL2



| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1011 | 0b001 |

Accessibility

MRS <Xt>, ICH\_VTR\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_VTR_EL2;
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_VTR_EL2;
```

A.8.10 ICH\_LR0\_EL2, Interrupt Controller List Registers

Provides interrupt context information for the virtual CPU interface.

Configurations

If EL2 is not implemented, this register is RES0 from EL3.

If list register n is not implemented, then accesses to this register are UNDEFINED.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

Functional group

GIC system registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-95: AArch64\_ich\_lr0\_el2 bit assignments

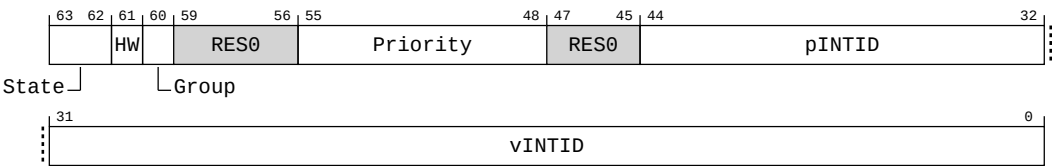


Table A-265: ICH\_LR0\_EL2 bit descriptions

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|---------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:62] | State | The state of the interrupt:<br><br><b>0b00</b><br>Invalid (Inactive).<br><br><b>0b01</b><br>Pending.<br><br><b>0b10</b><br>Active.<br><br><b>0b11</b><br>Pending and active.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | xx    |
| [61]    | HW    | Indicates whether this virtual interrupt maps directly to a hardware interrupt, meaning that it corresponds to a physical interrupt. Deactivation of the virtual interrupt also causes the deactivation of the physical interrupt with the ID that the pINTID field indicates.<br><br><b>0b0</b><br>The interrupt is triggered entirely by software. No notification is sent to the Distributor when the virtual interrupt is deactivated.<br><br><b>0b1</b><br>The interrupt maps directly to a hardware interrupt. A deactivate interrupt request is sent to the Distributor when the virtual interrupt is deactivated, using the pINTID field from this register to indicate the physical interrupt ID.<br><br>If AArch64-ICH_VMCR_EL2.VEOIM is 0, this request corresponds to a write to AArch64-ICC_EOIRO_EL1 or AArch64-ICC_EOIR1_EL1. Otherwise, it corresponds to a write to AArch64-ICC_DIR_EL1. | x     |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset  |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [60]    | Group    | <p>Indicates the group for this virtual interrupt.</p> <p><b>0b0</b></p> <p>This is a Group 0 virtual interrupt. AArch64-ICH_VMCR_EL2.VFIQEn determines whether it is signaled as a virtual IRQ or as a virtual FIQ, and AArch64-ICH_VMCR_EL2.VENG0 enables signaling of this interrupt to the virtual machine.</p> <p><b>0b1</b></p> <p>This is a Group 1 virtual interrupt, signaled as a virtual IRQ. AArch64-ICH_VMCR_EL2.VENG1 enables the signalling of this interrupt to the virtual machine.</p> <p>If AArch64-ICH_VMCR_EL2.VCBPR is 0, then AArch64-ICC_BPR1_EL1 determines if a pending Group 1 interrupt has sufficient priority to preempt current execution. Otherwise, AArch64-ICH_LR&lt;n&gt;_EL2 determines preemption.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | x      |
| [59:56] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0   |
| [55:48] | Priority | <p>The priority of this interrupt.</p> <p>It is IMPLEMENTATION DEFINED how many bits of priority are implemented, though at least five bits must be implemented. Unimplemented bits are <b>RES0</b> and start from bit[48] up to bit[50]. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.PRlbits.</p> <p>When ICH_LR&lt;n&gt;_EL2.NMI is set to 1, this field is <b>RES0</b> and the virtual interrupt's priority is treated as 0x00.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 8 {x}  |
| [47:45] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0   |
| [44:32] | pINTID   | <p>Physical INTID, for hardware interrupts.</p> <p>When ICH_LR&lt;n&gt;_EL2.HW is 0 (there is no corresponding physical interrupt), this field has the following meaning:</p> <ul style="list-style-type: none"> <li>Bits[44:42] : <b>RES0</b>.</li> <li>Bit[41] : EOI. If this bit is 1, then when the interrupt identified by vINTID is deactivated, a maintenance interrupt is asserted.</li> <li>Bits[40:32] : <b>RES0</b>.</li> </ul> <p>When ICH_LR&lt;n&gt;_EL2.HW is 1 (there is a corresponding physical interrupt):</p> <ul style="list-style-type: none"> <li>This field indicates the physical INTID. This field is only required to implement enough bits to hold a valid value for the implemented INTID size. Any unused higher order bits are <b>RES0</b>.</li> <li>When AArch64-ICC_CTLR_EL1.ExtRange is 0, then bits[44:42] of this field are <b>RES0</b>.</li> <li>If the value of pINTID is not a valid INTID, behavior is UNPREDICTABLE. If the value of pINTID indicates a PPI, this field applies to the PPI associated with this same physical PE ID as the virtual CPU interface requesting the deactivation.</li> </ul> <p>A hardware physical identifier is only required in List Registers for interrupts that require deactivation. This means only 13 bits of Physical INTID are required, regardless of the number specified by AArch64-ICC_CTLR_EL1.IDbits.</p> | 13 {x} |

| Bits   | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset  |
|--------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:0] | vINTID | <p>Virtual INTID of the interrupt.</p> <p>If the value of vINTID is 1020-1023 and ICH_LR&lt;n&gt;_EL2.State!=0b00 (Inactive), behavior is UNPREDICTABLE.</p> <p>Behavior is UNPREDICTABLE if two or more List Registers specify the same vINTID when:</p> <ul style="list-style-type: none"> <li>• ICH_LR&lt;n&gt;_EL2.State == 0b01.</li> <li>• ICH_LR&lt;n&gt;_EL2.State == 0b10.</li> <li>• ICH_LR&lt;n&gt;_EL2.State == 0b11.</li> </ul> <p>It is IMPLEMENTATION DEFINED how many bits are implemented, though at least 16 bits must be implemented. Unimplemented bits are <b>RES0</b>. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.IDbits.</p> <p>When AArch64-ICC_SRE_EL1.SRE == 0, specifying a vINTID in the LPI range is UNPREDICTABLE</p> <p><b>Note:</b><br/>When a VM is using memory-mapped access to the GIC, software must ensure that the correct source PE ID is provided in bits[12:10].</p> | 32 {x} |

## Access

MRS <Xt>, ICH\_LR0\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b000 |

MSR ICH\_LR0\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b000 |

## Accessibility

MRS <Xt>, ICH\_LR0\_EL2

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[0];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[0];

```

MSR ICH\_LR0\_EL2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_LR_EL2[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_LR_EL2[0] = X[t, 64];
```

A.8.11 ICH\_LR1\_EL2, Interrupt Controller List Registers

Provides interrupt context information for the virtual CPU interface.

Configurations

If EL2 is not implemented, this register is RES0 from EL3.

If list register n is not implemented, then accesses to this register are UNDEFINED.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

Functional group

GIC system registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 0    |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-96: AArch64\_ich\_lr1\_el2 bit assignments**

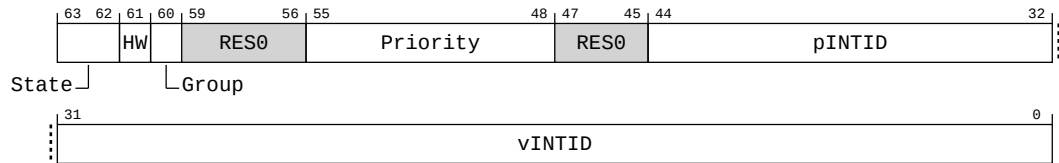


**Table A-268: ICH\_LR1\_EL2 bit descriptions**

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|---------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:62] | State | <p>The state of the interrupt:</p> <p><b>0b00</b><br/>Invalid (Inactive).</p> <p><b>0b01</b><br/>Pending.</p> <p><b>0b10</b><br/>Active.</p> <p><b>0b11</b><br/>Pending and active.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | xx    |
| [61]    | HW    | <p>Indicates whether this virtual interrupt maps directly to a hardware interrupt, meaning that it corresponds to a physical interrupt. Deactivation of the virtual interrupt also causes the deactivation of the physical interrupt with the ID that the pINTID field indicates.</p> <p><b>0b0</b><br/>The interrupt is triggered entirely by software. No notification is sent to the Distributor when the virtual interrupt is deactivated.</p> <p><b>0b1</b><br/>The interrupt maps directly to a hardware interrupt. A deactivate interrupt request is sent to the Distributor when the virtual interrupt is deactivated, using the pINTID field from this register to indicate the physical interrupt ID.</p> <p>If AArch64-ICH_VMCR_EL2.VEOIM is 0, this request corresponds to a write to AArch64-ICC_EOIRO_EL1 or AArch64-ICC_EOIR1_EL1. Otherwise, it corresponds to a write to AArch64-ICC_DIR_EL1.</p> | x     |
| [60]    | Group | <p>Indicates the group for this virtual interrupt.</p> <p><b>0b0</b><br/>This is a Group 0 virtual interrupt. AArch64-ICH_VMCR_EL2.VFIQEn determines whether it is signaled as a virtual IRQ or as a virtual FIQ, and AArch64-ICH_VMCR_EL2.VENG0 enables signaling of this interrupt to the virtual machine.</p> <p><b>0b1</b><br/>This is a Group 1 virtual interrupt, signaled as a virtual IRQ. AArch64-ICH_VMCR_EL2.VENG1 enables the signalling of this interrupt to the virtual machine.</p> <p>If AArch64-ICH_VMCR_EL2.VCBPR is 0, then AArch64-ICC_BPR1_EL1 determines if a pending Group 1 interrupt has sufficient priority to preempt current execution. Otherwise, AArch64-ICH_LR&lt;n&gt;_EL2 determines preemption.</p>                                                                                                                                                                              | x     |
| [59:56] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RES0  |

| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset       |
|---------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [55:48] | Priority    | <p>The priority of this interrupt.</p> <p>It is IMPLEMENTATION DEFINED how many bits of priority are implemented, though at least five bits must be implemented. Unimplemented bits are <b>RES0</b> and start from bit[48] up to bit[50]. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.PRIBits.</p> <p>When ICH_LR&lt;n&gt;_EL2.NMI is set to 1, this field is <b>RES0</b> and the virtual interrupt's priority is treated as 0x00.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 8 {x}       |
| [47:45] | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>RES0</b> |
| [44:32] | pINTID      | <p>Physical INTID, for hardware interrupts.</p> <p>When ICH_LR&lt;n&gt;_EL2.HW is 0 (there is no corresponding physical interrupt), this field has the following meaning:</p> <ul style="list-style-type: none"> <li>Bits[44:42] : <b>RES0</b>.</li> <li>Bit[41] : EOI. If this bit is 1, then when the interrupt identified by vINTID is deactivated, a maintenance interrupt is asserted.</li> <li>Bits[40:32] : <b>RES0</b>.</li> </ul> <p>When ICH_LR&lt;n&gt;_EL2.HW is 1 (there is a corresponding physical interrupt):</p> <ul style="list-style-type: none"> <li>This field indicates the physical INTID. This field is only required to implement enough bits to hold a valid value for the implemented INTID size. Any unused higher order bits are <b>RES0</b>.</li> <li>When AArch64-ICC_CTLR_EL1.ExtRange is 0, then bits[44:42] of this field are <b>RES0</b>.</li> <li>If the value of pINTID is not a valid INTID, behavior is UNPREDICTABLE. If the value of pINTID indicates a PPI, this field applies to the PPI associated with this same physical PE ID as the virtual CPU interface requesting the deactivation.</li> </ul> <p>A hardware physical identifier is only required in List Registers for interrupts that require deactivation. This means only 13 bits of Physical INTID are required, regardless of the number specified by AArch64-ICC_CTLR_EL1.IDbits.</p> | 13 {x}      |
| [31:0]  | vINTID      | <p>Virtual INTID of the interrupt.</p> <p>If the value of vINTID is 1020-1023 and ICH_LR&lt;n&gt;_EL2.State!=0b00 (Inactive), behavior is UNPREDICTABLE.</p> <p>Behavior is UNPREDICTABLE if two or more List Registers specify the same vINTID when:</p> <ul style="list-style-type: none"> <li>ICH_LR&lt;n&gt;_EL2.State == 0b01.</li> <li>ICH_LR&lt;n&gt;_EL2.State == 0b10.</li> <li>ICH_LR&lt;n&gt;_EL2.State == 0b11.</li> </ul> <p>It is IMPLEMENTATION DEFINED how many bits are implemented, though at least 16 bits must be implemented. Unimplemented bits are <b>RES0</b>. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.IDbits.</p> <p>When AArch64-ICC_SRE_EL1.SRE == 0, specifying a vINTID in the LPI range is UNPREDICTABLE</p> <p><b>Note:</b><br/>When a VM is using memory-mapped access to the GIC, software must ensure that the correct source PE ID is provided in bits[12:10].</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 32 {x}      |

## Access

MRS <Xt>, ICH\_LR1\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b001 |

MSR ICH\_LR1\_EL2, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b001 |

## Accessibility

MRS &lt;Xt&gt;, ICH\_LR1\_EL2

```

if 1 >= NUM_GIC_LIST_REGS then
    UNDEFINED;
elseif PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[1];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[1];

```

MSR ICH\_LR1\_EL2, &lt;Xt&gt;

```

if 1 >= NUM_GIC_LIST_REGS then
    UNDEFINED;
elseif PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_LR_EL2[1] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_LR_EL2[1] = X[t, 64];

```

## A.8.12 ICH\_LR2\_EL2, Interrupt Controller List Registers

Provides interrupt context information for the virtual CPU interface.

### Configurations

If EL2 is not implemented, this register is RES0 from EL3.

If list register n is not implemented, then accesses to this register are UNDEFINED.



This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

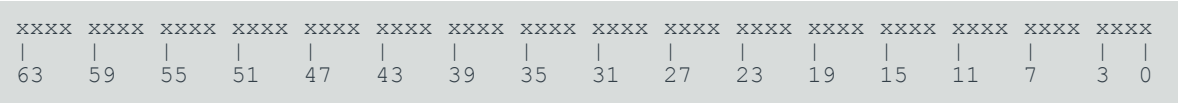
Functional group

GIC system registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-97: AArch64\_ich\_lr2\_el2 bit assignments

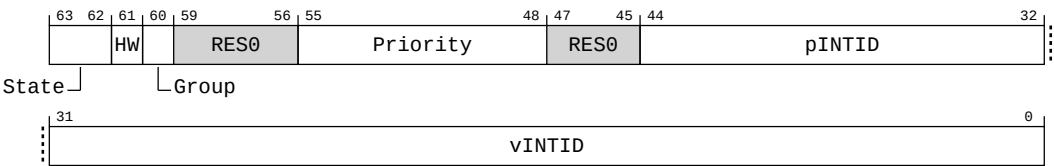


Table A-271: ICH\_LR2\_EL2 bit descriptions

| Bits    | Name  | Description                                                                                                                                                                  | Reset |
|---------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:62] | State | The state of the interrupt:<br><br><b>0b00</b><br>Invalid (Inactive).<br><br><b>0b01</b><br>Pending.<br><br><b>0b10</b><br>Active.<br><br><b>0b11</b><br>Pending and active. | xx    |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|---------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [61]    | HW       | <p>Indicates whether this virtual interrupt maps directly to a hardware interrupt, meaning that it corresponds to a physical interrupt. Deactivation of the virtual interrupt also causes the deactivation of the physical interrupt with the ID that the pINTID field indicates.</p> <p><b>0b0</b></p> <p>The interrupt is triggered entirely by software. No notification is sent to the Distributor when the virtual interrupt is deactivated.</p> <p><b>0b1</b></p> <p>The interrupt maps directly to a hardware interrupt. A deactivate interrupt request is sent to the Distributor when the virtual interrupt is deactivated, using the pINTID field from this register to indicate the physical interrupt ID.</p> <p>If AArch64-ICH_VMCR_EL2.VEOIM is 0, this request corresponds to a write to AArch64-ICC_EOIR0_EL1 or AArch64-ICC_EOIR1_EL1. Otherwise, it corresponds to a write to AArch64-ICC_DIR_EL1.</p> | x     |
| [60]    | Group    | <p>Indicates the group for this virtual interrupt.</p> <p><b>0b0</b></p> <p>This is a Group 0 virtual interrupt. AArch64-ICH_VMCR_EL2.VFIQEn determines whether it is signaled as a virtual IRQ or as a virtual FIQ, and AArch64-ICH_VMCR_EL2.VENG0 enables signaling of this interrupt to the virtual machine.</p> <p><b>0b1</b></p> <p>This is a Group 1 virtual interrupt, signaled as a virtual IRQ. AArch64-ICH_VMCR_EL2.VENG1 enables the signalling of this interrupt to the virtual machine.</p> <p>If AArch64-ICH_VMCR_EL2.VCBPR is 0, then AArch64-ICC_BPR1_EL1 determines if a pending Group 1 interrupt has sufficient priority to preempt current execution. Otherwise, AArch64-ICH_LR&lt;n&gt;_EL2 determines preemption.</p>                                                                                                                                                                              | x     |
| [59:56] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [55:48] | Priority | <p>The priority of this interrupt.</p> <p>It is IMPLEMENTATION DEFINED how many bits of priority are implemented, though at least five bits must be implemented. Unimplemented bits are <b>RES0</b> and start from bit[48] up to bit[50]. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.PRIBits.</p> <p>When ICH_LR&lt;n&gt;_EL2.NMI is set to 1, this field is <b>RES0</b> and the virtual interrupt's priority is treated as 0x00.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 8 {x} |
| [47:45] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset  |
|---------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [44:32] | pINTID | <p>Physical INTID, for hardware interrupts.</p> <p>When ICH_LR&lt;n&gt;_EL2.HW is 0 (there is no corresponding physical interrupt), this field has the following meaning:</p> <ul style="list-style-type: none"> <li>Bits[44:42] : <b>RES0</b>.</li> <li>Bit[41] : EOI. If this bit is 1, then when the interrupt identified by vINTID is deactivated, a maintenance interrupt is asserted.</li> <li>Bits[40:32] : <b>RES0</b>.</li> </ul> <p>When ICH_LR&lt;n&gt;_EL2.HW is 1 (there is a corresponding physical interrupt):</p> <ul style="list-style-type: none"> <li>This field indicates the physical INTID. This field is only required to implement enough bits to hold a valid value for the implemented INTID size. Any unused higher order bits are <b>RES0</b>.</li> <li>When AArch64-ICC_CTLR_EL1.ExtRange is 0, then bits[44:42] of this field are <b>RES0</b>.</li> <li>If the value of pINTID is not a valid INTID, behavior is UNPREDICTABLE. If the value of pINTID indicates a PPI, this field applies to the PPI associated with this same physical PE ID as the virtual CPU interface requesting the deactivation.</li> </ul> <p>A hardware physical identifier is only required in List Registers for interrupts that require deactivation. This means only 13 bits of Physical INTID are required, regardless of the number specified by AArch64-ICC_CTLR_EL1.IDbits.</p> | 13 {x} |
| [31:0]  | vINTID | <p>Virtual INTID of the interrupt.</p> <p>If the value of vINTID is 1020-1023 and ICH_LR&lt;n&gt;_EL2.State!=0b00 (Inactive), behavior is UNPREDICTABLE.</p> <p>Behavior is UNPREDICTABLE if two or more List Registers specify the same vINTID when:</p> <ul style="list-style-type: none"> <li>ICH_LR&lt;n&gt;_EL2.State == 0b01.</li> <li>ICH_LR&lt;n&gt;_EL2.State == 0b10.</li> <li>ICH_LR&lt;n&gt;_EL2.State == 0b11.</li> </ul> <p>It is IMPLEMENTATION DEFINED how many bits are implemented, though at least 16 bits must be implemented. Unimplemented bits are <b>RES0</b>. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.IDbits.</p> <p>When AArch64-ICC_SRE_EL1.SRE == 0, specifying a vINTID in the LPI range is UNPREDICTABLE</p> <p><b>Note:</b><br/>When a VM is using memory-mapped access to the GIC, software must ensure that the correct source PE ID is provided in bits[12:10].</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 32 {x} |

## Access

MRS <Xt>, ICH\_LR2\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b010 |

MSR ICH\_LR2\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b010 |

## Accessibility

MRS <Xt>, ICH\_LR2\_EL2

```

if 2 >= NUM_GIC_LIST_REGS then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[2];
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[2];

```

MSR ICH\_LR2\_EL2, <Xt>

```

if 2 >= NUM_GIC_LIST_REGS then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_LR_EL2[2] = X[t, 64];
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_LR_EL2[2] = X[t, 64];

```

## A.8.13 ICH\_LR3\_EL2, Interrupt Controller List Registers

Provides interrupt context information for the virtual CPU interface.

### Configurations

If EL2 is not implemented, this register is RES0 from EL3.

If list register *n* is not implemented, then accesses to this register are UNDEFINED.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

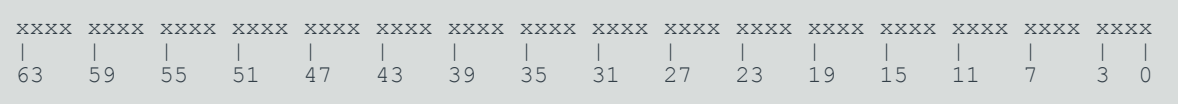
Functional group

GIC system registers

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-98: AArch64\_ich\_lr3\_el2 bit assignments

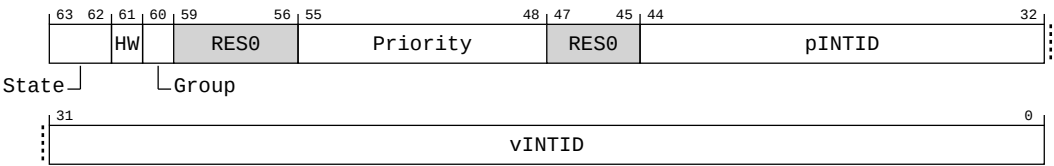


Table A-274: ICH\_LR3\_EL2 bit descriptions

| Bits    | Name  | Description                                                                                                                                                                  | Reset |
|---------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:62] | State | The state of the interrupt:<br><br><b>0b00</b><br>Invalid (Inactive).<br><br><b>0b01</b><br>Pending.<br><br><b>0b10</b><br>Active.<br><br><b>0b11</b><br>Pending and active. | xx    |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|---------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [61]    | HW       | <p>Indicates whether this virtual interrupt maps directly to a hardware interrupt, meaning that it corresponds to a physical interrupt. Deactivation of the virtual interrupt also causes the deactivation of the physical interrupt with the ID that the pINTID field indicates.</p> <p><b>0b0</b></p> <p>The interrupt is triggered entirely by software. No notification is sent to the Distributor when the virtual interrupt is deactivated.</p> <p><b>0b1</b></p> <p>The interrupt maps directly to a hardware interrupt. A deactivate interrupt request is sent to the Distributor when the virtual interrupt is deactivated, using the pINTID field from this register to indicate the physical interrupt ID.</p> <p>If AArch64-ICH_VMCR_EL2.VEOIM is 0, this request corresponds to a write to AArch64-ICC_EOIR0_EL1 or AArch64-ICC_EOIR1_EL1. Otherwise, it corresponds to a write to AArch64-ICC_DIR_EL1.</p> | x     |
| [60]    | Group    | <p>Indicates the group for this virtual interrupt.</p> <p><b>0b0</b></p> <p>This is a Group 0 virtual interrupt. AArch64-ICH_VMCR_EL2.VFIQEn determines whether it is signaled as a virtual IRQ or as a virtual FIQ, and AArch64-ICH_VMCR_EL2.VENG0 enables signaling of this interrupt to the virtual machine.</p> <p><b>0b1</b></p> <p>This is a Group 1 virtual interrupt, signaled as a virtual IRQ. AArch64-ICH_VMCR_EL2.VENG1 enables the signalling of this interrupt to the virtual machine.</p> <p>If AArch64-ICH_VMCR_EL2.VCBPR is 0, then AArch64-ICC_BPR1_EL1 determines if a pending Group 1 interrupt has sufficient priority to preempt current execution. Otherwise, AArch64-ICH_LR&lt;n&gt;_EL2 determines preemption.</p>                                                                                                                                                                              | x     |
| [59:56] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [55:48] | Priority | <p>The priority of this interrupt.</p> <p>It is IMPLEMENTATION DEFINED how many bits of priority are implemented, though at least five bits must be implemented. Unimplemented bits are <b>RES0</b> and start from bit[48] up to bit[50]. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.PRIBits.</p> <p>When ICH_LR&lt;n&gt;_EL2.NMI is set to 1, this field is <b>RES0</b> and the virtual interrupt's priority is treated as 0x00.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 8 {x} |
| [47:45] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset  |
|---------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [44:32] | pINTID | <p>Physical INTID, for hardware interrupts.</p> <p>When ICH_LR&lt;n&gt;_EL2.HW is 0 (there is no corresponding physical interrupt), this field has the following meaning:</p> <ul style="list-style-type: none"> <li>Bits[44:42] : <b>RES0</b>.</li> <li>Bit[41] : EOI. If this bit is 1, then when the interrupt identified by vINTID is deactivated, a maintenance interrupt is asserted.</li> <li>Bits[40:32] : <b>RES0</b>.</li> </ul> <p>When ICH_LR&lt;n&gt;_EL2.HW is 1 (there is a corresponding physical interrupt):</p> <ul style="list-style-type: none"> <li>This field indicates the physical INTID. This field is only required to implement enough bits to hold a valid value for the implemented INTID size. Any unused higher order bits are <b>RES0</b>.</li> <li>When AArch64-ICC_CTLR_EL1.ExtRange is 0, then bits[44:42] of this field are <b>RES0</b>.</li> <li>If the value of pINTID is not a valid INTID, behavior is UNPREDICTABLE. If the value of pINTID indicates a PPI, this field applies to the PPI associated with this same physical PE ID as the virtual CPU interface requesting the deactivation.</li> </ul> <p>A hardware physical identifier is only required in List Registers for interrupts that require deactivation. This means only 13 bits of Physical INTID are required, regardless of the number specified by AArch64-ICC_CTLR_EL1.IDbits.</p> | 13 {x} |
| [31:0]  | vINTID | <p>Virtual INTID of the interrupt.</p> <p>If the value of vINTID is 1020-1023 and ICH_LR&lt;n&gt;_EL2.State!=0b00 (Inactive), behavior is UNPREDICTABLE.</p> <p>Behavior is UNPREDICTABLE if two or more List Registers specify the same vINTID when:</p> <ul style="list-style-type: none"> <li>ICH_LR&lt;n&gt;_EL2.State == 0b01.</li> <li>ICH_LR&lt;n&gt;_EL2.State == 0b10.</li> <li>ICH_LR&lt;n&gt;_EL2.State == 0b11.</li> </ul> <p>It is IMPLEMENTATION DEFINED how many bits are implemented, though at least 16 bits must be implemented. Unimplemented bits are <b>RES0</b>. The number of implemented bits can be discovered from AArch64-ICH_VTR_EL2.IDbits.</p> <p>When AArch64-ICC_SRE_EL1.SRE == 0, specifying a vINTID in the LPI range is UNPREDICTABLE</p> <p><b>Note:</b><br/>When a VM is using memory-mapped access to the GIC, software must ensure that the correct source PE ID is provided in bits[12:10].</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 32 {x} |

## Access

MRS <Xt>, ICH\_LR3\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b011 |

MSR ICH\_LR3\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1100 | 0b1100 | 0b011 |

## Accessibility

MRS <Xt>, ICH\_LR3\_EL2

```

if 3 >= NUM_GIC_LIST_REGS then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[3];
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_LR_EL2[3];

```

MSR ICH\_LR3\_EL2, <Xt>

```

if 3 >= NUM_GIC_LIST_REGS then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_LR_EL2[3] = X[t, 64];
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_LR_EL2[3] = X[t, 64];

```

## A.8.14 ICC\_CTLR\_EL3, Interrupt Controller Control Register (EL3)

Controls aspects of the behavior of the GIC CPU interface and provides information about the features implemented.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64



**Functional group**

GIC system registers

**Access type**

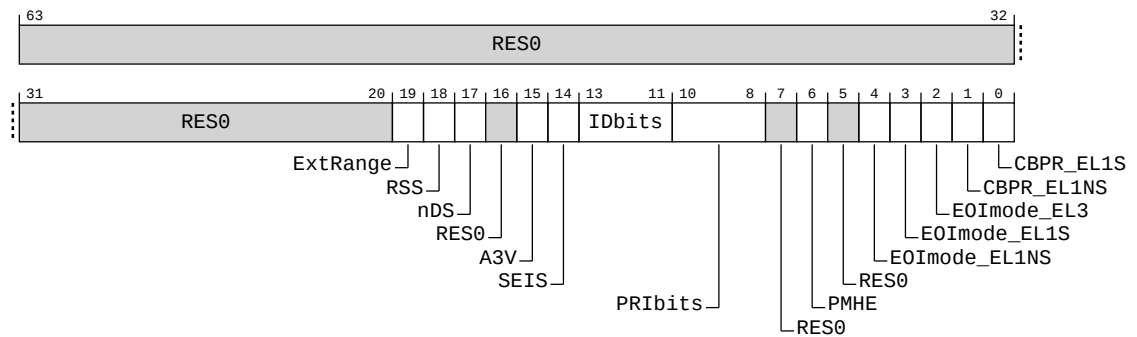
See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | x0xx | xxxx |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |

**Note**

Where the reset reads xxxx, see individual bits.

**Bit descriptions****Figure A-99: AArch64\_icc\_ctlr\_el3 bit assignments****Table A-277: ICC\_CTLR\_EL3 bit descriptions**

| Bits    | Name     | Description                                                                                                                                                                                                                             | Reset |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:20] | RES0     | Reserved                                                                                                                                                                                                                                | RES0  |
| [19]    | ExtRange | Extended INTID range (read-only).<br><b>0b1</b><br>CPU interface supports INTIDs in the range 1024..8191<br><ul style="list-style-type: none"> <li>All INTIDs in the range 1024..8191 are treated as requiring deactivation.</li> </ul> | x     |
| [18]    | RSS      | Range Selector Support.<br><b>0b0</b><br>Targeted SGIs with affinity level 0 values of 0-15 are supported.                                                                                                                              | x     |
| [17]    | nDS      | Disable Security not supported. Read-only and writes are ignored.<br><b>0b1</b><br>The CPU interface logic does not support disabling of security, and requires that security is not disabled.                                          | x     |
| [16]    | RES0     | Reserved                                                                                                                                                                                                                                | RES0  |

| Bits    | Name          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|---------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15]    | A3V           | Affinity 3 Valid. Read-only and writes are ignored.<br><br><b>0b1</b><br>The CPU interface logic supports nonzero values of the Aff3 field in SGI generation System registers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | x     |
| [14]    | SEIS          | SEI Support. Read-only and writes are ignored. Indicates whether the CPU interface supports generation of SEIs:<br><br><b>0b0</b><br>The CPU interface logic does not support generation of SEIs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | x     |
| [13:11] | IDbits        | Identifier bits. Read-only and writes are ignored. Indicates the number of physical interrupt identifier bits supported.<br><br><b>0b000</b><br>16 bits.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | xxx   |
| [10:8]  | PRIbits       | Priority bits. Read-only and writes are ignored. The number of priority bits implemented, minus one.<br><br>An implementation that supports two Security states must implement at least 32 levels of physical priority (5 priority bits).<br><br>An implementation that supports only a single Security state must implement at least 16 levels of physical priority (4 priority bits).<br><br><b>Note:</b><br>This field always returns the number of priority bits implemented, regardless of the value of SCR_EL3.NS or the value of ext-GICD_CTLR.DS.<br><br>The division between group priority and subpriority is defined in the binary point registers AArch64-ICC_BPR0_EL1 and AArch64-ICC_BPR1_EL1.<br><br>This field determines the minimum value of ICC_BPR0_EL1.<br><br><b>0b100</b><br>5 bits of priority are implemented | xxx   |
| [7]     | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RES0  |
| [6]     | PMHE          | Priority Mask Hint Enable.<br><br><b>0b0</b><br>Disables use of the priority mask register as a hint for interrupt distribution.<br><br><b>0b1</b><br>Enables use of the priority mask register as a hint for interrupt distribution.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0b0   |
| [5]     | RES0          | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RES0  |
| [4]     | EOImode_EL1NS | EOI mode for interrupts handled at Non-secure EL1 and EL2. Controls whether a write to an End of Interrupt register also deactivates the interrupt.<br><br><b>0b0</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide both priority drop and interrupt deactivation functionality. Accesses to AArch64-ICC_DIR_EL1 are <b>UNPREDICTABLE</b> .<br><br><b>0b1</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide priority drop functionality only. AArch64-ICC_DIR_EL1 provides interrupt deactivation functionality.                                                                                                                                                                                                                                                                                             | x     |

| Bits | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [3]  | EOImode_EL1S | EOI mode for interrupts handled at Secure EL1 and EL2. Controls whether a write to an End of Interrupt register also deactivates the interrupt.<br><br><b>0b0</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide both priority drop and interrupt deactivation functionality. Accesses to AArch64-ICC_DIR_EL1 are <b>UNPREDICTABLE</b> .<br><br><b>0b1</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide priority drop functionality only. AArch64-ICC_DIR_EL1 provides interrupt deactivation functionality.                                                                                               | x     |
| [2]  | EOImode_EL3  | EOI mode for interrupts handled at EL3. Controls whether a write to an End of Interrupt register also deactivates the interrupt.<br><br><b>0b0</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide both priority drop and interrupt deactivation functionality. Accesses to AArch64-ICC_DIR_EL1 are <b>UNPREDICTABLE</b> .<br><br><b>0b1</b><br>AArch64-ICC_EOIR0_EL1 and AArch64-ICC_EOIR1_EL1 provide priority drop functionality only. AArch64-ICC_DIR_EL1 provides interrupt deactivation functionality.                                                                                                              | x     |
| [1]  | CBPR_EL1NS   | Common Binary Point Register, EL1 Non-secure. Controls whether the same register is used for interrupt preemption of both Group 0 and Group 1 Non-secure interrupts at EL1 and EL2.<br><br><b>0b0</b><br>AArch64-ICC_BPR0_EL1 determines the preemption group for Group 0 interrupts only.<br><br>AArch64-ICC_BPR1_EL1 determines the preemption group for Non-secure Group 1 interrupts.<br><br><b>0b1</b><br>AArch64-ICC_BPR0_EL1 determines the preemption group for Group 0 interrupts and Non-secure Group 1 interrupts. Non-secure accesses to ext-GICC_BPR and AArch64-ICC_BPR1_EL1 access the state of AArch64-ICC_BPR0_EL1. | x     |
| [0]  | CBPR_EL1S    | Common Binary Point Register, EL1 Secure. Controls whether the same register is used for interrupt preemption of both Group 0 and Group 1 Secure interrupts at EL1 and EL2.<br><br><b>0b0</b><br>AArch64-ICC_BPR0_EL1 determines the preemption group for Group 0 interrupts only.<br><br>AArch64-ICC_BPR1_EL1 determines the preemption group for Secure Group 1 interrupts.<br><br><b>0b1</b><br>AArch64-ICC_BPR0_EL1 determines the preemption group for Group 0 interrupts and Secure Group 1 interrupts. Secure EL1 accesses to AArch64-ICC_BPR1_EL1 access the state of AArch64-ICC_BPR0_EL1.                                  | x     |

## Access

MRS <Xt>, ICC\_CTLR\_EL3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1100 | 0b1100 | 0b100 |

MSR ICC\_CTLR\_EL3, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b110 | 0b1100 | 0b1100 | 0b100 |

Accessibility

MRS <Xt>, ICC\_CTLR\_EL3

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_CTLR_EL3;
```

MSR ICC\_CTLR\_EL3, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    UNDEFINED;
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICC_CTLR_EL3 = X[t, 64];
```

A.9 AArch64 Generic Timer registers summary

The following summary table provides an overview of all Generic Timer registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

Table A-280: Generic Timer registers summary

| Name        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                           |
|-------------|-----|-----|-----|-----|-----|----------------------------|--------|---------------------------------------|
| CNTKCTL_EL1 | 3   | 0   | C14 | C1  | 0   | See individual bit resets. | 64-bit | Counter-timer Kernel Control Register |
| CNTFRQ_ELO  | 3   | 3   | C14 | C0  | 0   | See individual bit resets. | 64-bit | Counter-timer Frequency Register      |
| CNTPCT_ELO  | 3   | 3   | C14 | C0  | 1   | See individual bit resets. | 64-bit | Counter-timer Physical Count Register |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                    |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------------|
| CNTVCT_ELO      | 3   | 3   | C14 | C0  | 2   | See individual bit resets. | 64-bit | Counter-timer Virtual Count Register                           |
| CNTPCTSS_ELO    | 3   | 3   | C14 | C0  | 5   | See individual bit resets. | 64-bit | Counter-timer Self-Synchronized Physical Count Register        |
| CNTVCTSS_ELO    | 3   | 3   | C14 | C0  | 6   | See individual bit resets. | 64-bit | Counter-timer Self-Synchronized Virtual Count Register         |
| CNTP_TVAL_ELO   | 3   | 3   | C14 | C2  | 0   | See individual bit resets. | 64-bit | Counter-timer Physical Timer TimerValue Register               |
| CNTP_CTL_ELO    | 3   | 3   | C14 | C2  | 1   | See individual bit resets. | 64-bit | Counter-timer Physical Timer Control Register                  |
| CNTP_CVAL_ELO   | 3   | 3   | C14 | C2  | 2   | See individual bit resets. | 64-bit | Counter-timer Physical Timer CompareValue Register             |
| CNTV_TVAL_ELO   | 3   | 3   | C14 | C3  | 0   | See individual bit resets. | 64-bit | Counter-timer Virtual Timer TimerValue Register                |
| CNTV_CTL_ELO    | 3   | 3   | C14 | C3  | 1   | See individual bit resets. | 64-bit | Counter-timer Virtual Timer Control Register                   |
| CNTV_CVAL_ELO   | 3   | 3   | C14 | C3  | 2   | See individual bit resets. | 64-bit | Counter-timer Virtual Timer CompareValue Register              |
| CNTVOFF_EL2     | 3   | 4   | C14 | C0  | 3   | See individual bit resets. | 64-bit | Counter-timer Virtual Offset Register                          |
| CNTPOFF_EL2     | 3   | 4   | C14 | C0  | 6   | See individual bit resets. | 64-bit | Counter-timer Physical Offset Register                         |
| CNTHCTL_EL2     | 3   | 4   | C14 | C1  | 0   | See individual bit resets. | 64-bit | Counter-timer Hypervisor Control Register                      |
| CNTHP_TVAL_EL2  | 3   | 4   | C14 | C2  | 0   | See individual bit resets. | 64-bit | Counter-timer Physical Timer TimerValue Register (EL2)         |
| CNTHP_CTL_EL2   | 3   | 4   | C14 | C2  | 1   | See individual bit resets. | 64-bit | Counter-timer Hypervisor Physical Timer Control Register       |
| CNTHP_CVAL_EL2  | 3   | 4   | C14 | C2  | 2   | See individual bit resets. | 64-bit | Counter-timer Physical Timer CompareValue Register (EL2)       |
| CNTHV_TVAL_EL2  | 3   | 4   | C14 | C3  | 0   | See individual bit resets. | 64-bit | Counter-timer Virtual Timer TimerValue Register (EL2)          |
| CNTHV_CTL_EL2   | 3   | 4   | C14 | C3  | 1   | See individual bit resets. | 64-bit | Counter-timer Virtual Timer Control Register (EL2)             |
| CNTHV_CVAL_EL2  | 3   | 4   | C14 | C3  | 2   | See individual bit resets. | 64-bit | Counter-timer Virtual Timer CompareValue Register (EL2)        |
| CNTHVS_TVAL_EL2 | 3   | 4   | C14 | C4  | 0   | See individual bit resets. | 64-bit | Counter-timer Secure Virtual Timer TimerValue Register (EL2)   |
| CNTHVS_CTL_EL2  | 3   | 4   | C14 | C4  | 1   | See individual bit resets. | 64-bit | Counter-timer Secure Virtual Timer Control Register (EL2)      |
| CNTHVS_CVAL_EL2 | 3   | 4   | C14 | C4  | 2   | See individual bit resets. | 64-bit | Counter-timer Secure Virtual Timer CompareValue Register (EL2) |
| CNTHPS_TVAL_EL2 | 3   | 4   | C14 | C5  | 0   | See individual bit resets. | 64-bit | Counter-timer Secure Physical Timer TimerValue Register (EL2)  |
| CNTHPS_CTL_EL2  | 3   | 4   | C14 | C5  | 1   | See individual bit resets. | 64-bit | Counter-timer Secure Physical Timer Control Register (EL2)     |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                     |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------------------------|
| CNTHPS_CVAL_EL2 | 3   | 4   | C14 | C5  | 2   | See individual bit resets. | 64-bit | Counter-timer Secure Physical Timer CompareValue Register (EL2) |
| CNTPS_TVAL_EL1  | 3   | 7   | C14 | C2  | 0   | See individual bit resets. | 64-bit | Counter-timer Physical Secure Timer TimerValue Register         |
| CNTPS_CTL_EL1   | 3   | 7   | C14 | C2  | 1   | See individual bit resets. | 64-bit | Counter-timer Physical Secure Timer Control Register            |
| CNTPS_CVAL_EL1  | 3   | 7   | C14 | C2  | 2   | See individual bit resets. | 64-bit | Counter-timer Physical Secure Timer CompareValue Register       |

## A.10 AArch64 Other system control registers summary

The following summary table provides an overview of all Other system control registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-281: Other system control registers summary**

| Name       | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                       |
|------------|-----|-----|-----|-----|-----|----------------------------|--------|---------------------------------------------------|
| SCTLR_EL1  | 3   | 0   | C1  | C0  | 0   | See individual bit resets. | 64-bit | System Control Register (EL1)                     |
| CPACR_EL1  | 3   | 0   | C1  | C0  | 2   | See individual bit resets. | 64-bit | Architectural Feature Access Control Register     |
| ZCR_EL1    | 3   | 0   | C1  | C2  | 0   | See individual bit resets. | 64-bit | SVE Control Register (EL1)                        |
| SCTLR_EL2  | 3   | 4   | C1  | C0  | 0   | See individual bit resets. | 64-bit | System Control Register (EL2)                     |
| HCR_EL2    | 3   | 4   | C1  | C1  | 0   | See individual bit resets. | 64-bit | Hypervisor Configuration Register                 |
| CPTR_EL2   | 3   | 4   | C1  | C1  | 2   | See individual bit resets. | 64-bit | Architectural Feature Trap Register (EL2)         |
| HSTR_EL2   | 3   | 4   | C1  | C1  | 3   | See individual bit resets. | 64-bit | Hypervisor System Trap Register                   |
| HFGRTR_EL2 | 3   | 4   | C1  | C1  | 4   | See individual bit resets. | 64-bit | Hypervisor Fine-Grained Read Trap Register        |
| HFGWTR_EL2 | 3   | 4   | C1  | C1  | 5   | See individual bit resets. | 64-bit | Hypervisor Fine-Grained Write Trap Register       |
| HFGITR_EL2 | 3   | 4   | C1  | C1  | 6   | See individual bit resets. | 64-bit | Hypervisor Fine-Grained Instruction Trap Register |

| Name        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                                  |
|-------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------------------|
| ZCR_EL2     | 3   | 4   | C1  | C2  | 0   | See individual bit resets. | 64-bit | SVE Control Register (EL2)                                   |
| HCRX_EL2    | 3   | 4   | C1  | C2  | 2   | See individual bit resets. | 64-bit | Extended Hypervisor Configuration Register                   |
| HDFGRTR_EL2 | 3   | 4   | C3  | C1  | 4   | See individual bit resets. | 64-bit | Hypervisor Debug Fine-Grained Read Trap Register             |
| HDFGWTR_EL2 | 3   | 4   | C3  | C1  | 5   | See individual bit resets. | 64-bit | Hypervisor Debug Fine-Grained Write Trap Register            |
| HAFGRTR_EL2 | 3   | 4   | C3  | C1  | 6   | See individual bit resets. | 64-bit | Hypervisor Activity Monitors Fine-Grained Read Trap Register |
| SCTLR_EL3   | 3   | 6   | C1  | C0  | 0   | See individual bit resets. | 64-bit | System Control Register (EL3)                                |
| ZCR_EL3     | 3   | 6   | C1  | C2  | 0   | See individual bit resets. | 64-bit | SVE Control Register (EL3)                                   |

## A.11 AArch64 Activity Monitors registers summary

The following summary table provides an overview of all Activity Monitors registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-282: Activity Monitors registers summary**

| Name                       | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                            |
|----------------------------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------------------|
| AMCR_ELO                   | 3   | 3   | C13 | C2  | 0   | See individual bit resets. | 64-bit | Activity Monitors Control Register                     |
| <a href="#">AMCFGR_ELO</a> | 3   | 3   | C13 | C2  | 1   | See individual bit resets. | 64-bit | Activity Monitors Configuration Register               |
| <a href="#">AMCGCR_ELO</a> | 3   | 3   | C13 | C2  | 2   | See individual bit resets. | 64-bit | Activity Monitors Counter Group Configuration Register |
| AMUSERENR_ELO              | 3   | 3   | C13 | C2  | 3   | See individual bit resets. | 64-bit | Activity Monitors User Enable Register                 |
| AMCNTENCLR0_ELO            | 3   | 3   | C13 | C2  | 4   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Clear Register 0        |
| AMCNTENSET0_ELO            | 3   | 3   | C13 | C2  | 5   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Set Register 0          |
| AMCNTENCLR1_ELO            | 3   | 3   | C13 | C3  | 0   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Clear Register 1        |

| Name            | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                   |
|-----------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------|
| AMCNTENSET1_ELO | 3   | 3   | C13 | C3  | 1   | See individual bit resets. | 64-bit | Activity Monitors Count Enable Set Register 1 |
| AMEVCNTR00_ELO  | 3   | 3   | C13 | C4  | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0   |
| AMEVCNTR01_ELO  | 3   | 3   | C13 | C4  | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0   |
| AMEVCNTR02_ELO  | 3   | 3   | C13 | C4  | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0   |
| AMEVCNTR03_ELO  | 3   | 3   | C13 | C4  | 3   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 0   |
| AMEVTYPER00_ELO | 3   | 3   | C13 | C6  | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0      |
| AMEVTYPER01_ELO | 3   | 3   | C13 | C6  | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0      |
| AMEVTYPER02_ELO | 3   | 3   | C13 | C6  | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0      |
| AMEVTYPER03_ELO | 3   | 3   | C13 | C6  | 3   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 0      |
| AMEVCNTR10_ELO  | 3   | 3   | C13 | C12 | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 1   |
| AMEVCNTR11_ELO  | 3   | 3   | C13 | C12 | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 1   |
| AMEVCNTR12_ELO  | 3   | 3   | C13 | C12 | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Counter Registers 1   |
| AMEVTYPER10_ELO | 3   | 3   | C13 | C14 | 0   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 1      |
| AMEVTYPER11_ELO | 3   | 3   | C13 | C14 | 1   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 1      |
| AMEVTYPER12_ELO | 3   | 3   | C13 | C14 | 2   | See individual bit resets. | 64-bit | Activity Monitors Event Type Registers 1      |

### A.11.1 AMCFGR\_ELO, Activity Monitors Configuration Register

Global configuration register for the activity monitors.

Provides information on supported features, the number of counter groups implemented, the total number of activity monitor event counters implemented, and the size of the counters. AMCFGR\_ELO is applicable to both the architected and the auxiliary counter groups.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64



Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0001 | xxx1 | 0000 | 0000 | 0011 | 1111 | 0000 | 0110 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-100: AArch64\_amcfr\_el0 bit assignments

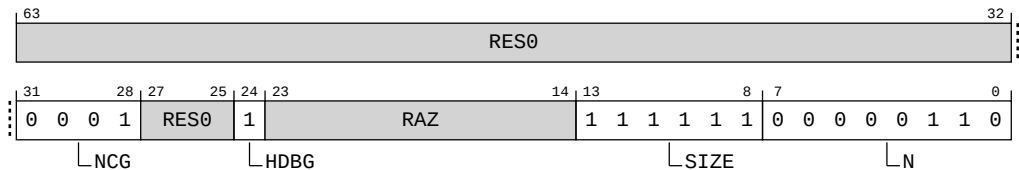


Table A-283: AMCFGR\_EL0 bit descriptions

| Bits    | Name | Description                                                                                                                                     | Reset  |
|---------|------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:32] | RES0 | Reserved                                                                                                                                        | RES0   |
| [31:28] | NCG  | Defines the number of counter groups. The following value is specified for this product.<br><b>0b0001</b><br>Two counter groups are implemented | 0b0001 |
| [27:25] | RES0 | Reserved                                                                                                                                        | RES0   |
| [24]    | HDBG | Halt-on-debug supported.<br><br>This feature must be supported, and so this bit is 0b1.<br><b>0b1</b><br>AArch64-AMCR_EL0.HDBG is read/write.   | 0b1    |
| [23:14] | RAZ  | Reserved                                                                                                                                        | RAZ    |

| Bits   | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                 | Reset    |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [13:8] | SIZE | <p>Defines the size of activity monitor event counters.</p> <p>The size of the activity monitor event counters implemented by the activity monitors Extension is [AMCFGR_ELO.SIZE + 1].</p> <p><b>Note:</b><br/>Software also uses this field to determine the spacing of counters in the memory-map. From Armv8, the counters are at doubleword-aligned addresses.</p> <p><b>0b111111</b><br/>64 bits.</p> | 0b111111 |
| [7:0]  | N    | <p>Defines the number of activity monitor event counters.</p> <p>The total number of counters implemented in all groups by the Activity Monitors Extension is [AMCFGR_ELO.N + 1].</p> <p><b>0b00000110</b><br/>Seven activity monitor event counters</p>                                                                                                                                                    | 0x06     |

## Access

MRS <Xt>, AMCFGR\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0010 | 0b001 |

## Accessibility

MRS <Xt>, AMCFGR\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCFGR_ELO;
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCFGR_ELO;
        elsif PSTATE.EL == EL2 then
            if CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;

```

```
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMCFGR_EL0;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = AMCFGR_EL0;
```

### A.11.2 AMCGCR\_EL0, Activity Monitors Counter Group Configuration Register

Provides information on the number of activity monitor event counters implemented within each counter group.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Activity Monitors registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0011 | 0000 | 0100 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-101: AArch64\_amcgcr\_el0 bit assignments

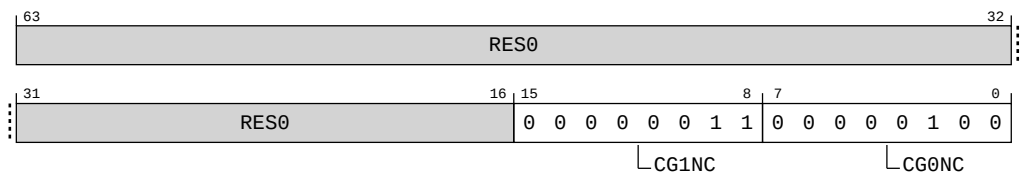


**Table A-285: AMCGCR\_EL0 bit descriptions**

| Bits    | Name  | Description                                                                                                                                         | Reset |
|---------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:16] | RES0  | Reserved                                                                                                                                            | RES0  |
| [15:8]  | CG1NC | Counter Group 1 Number of Counters. The number of counters in the auxiliary counter group.<br><b>0b00000011</b><br>The permitted range of counters. | 0x03  |
| [7:0]   | CG0NC | Counter Group 0 Number of Counters. The number of counters in the architected counter group.<br><b>0b00000100</b>                                   | 0x04  |

### Access

MRS &lt;Xt&gt;, AMCGCR\_EL0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0010 | 0b010 |

### Accessibility

MRS &lt;Xt&gt;, AMCGCR\_EL0

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCGCR_EL0;
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCGCR_EL0;
        elsif PSTATE.EL == EL2 then
            if CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCGCR_EL0;
        elsif PSTATE.EL == EL3 then
            X[t, 64] = AMCGCR_EL0;

```

### A.11.3 AMEVCNTR00\_ELO, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counter 0.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Activity Monitors registers

##### Access type

See bit descriptions

##### Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

#### Bit descriptions

Figure A-102: AArch64\_amevcntr00\_el0 bit assignments

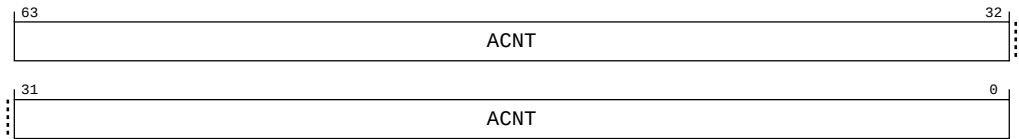


Table A-287: AMEVCNTR00\_ELO bit descriptions

| Bits   | Name | Description                                            | Reset              |
|--------|------|--------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of architected activity monitor event counter 0. | 0x0000000000000000 |

#### Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR00\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b000 |

MSR AMEVCNTR00\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b000 |

## Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are UNDEFINED.



Note

AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR00\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
            HAFGRTR_EL2.AMEVCNTR00_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR0_ELO[0];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVCNTR00_ELO == '1'
            then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    X[t, 64] = AMEVCNTR0_ELO[0];
        elsif PSTATE.EL == EL2 then
            if CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR0_ELO[0];
        elsif PSTATE.EL == EL3 then

```

```
X[t, 64] = AMEVCNTR0_ELO[0];
```

MSR AMEVCNTR00\_ELO, <Xt>

```
if IsHighestEL(PSTATE.EL) then
    AMEVCNTR0_ELO[0] = X[t, 64];
else
    UNDEFINED;
```

A.11.4 AMEVCNTR01\_ELO, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counter 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure A-103: AArch64\_amevcntr01\_el0 bit assignments

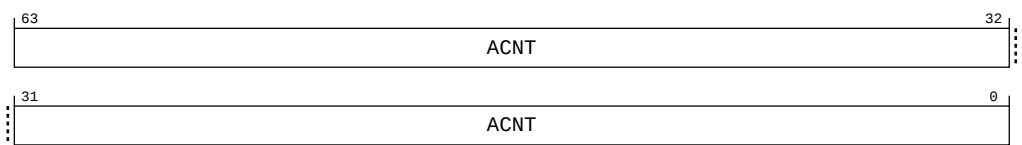


Table A-290: AMEVCNTR01\_ELO bit descriptions

| Bits   | Name | Description                                            | Reset              |
|--------|------|--------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of architected activity monitor event counter 1. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are **UNDEFINED**.

**Note**

AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR01\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b001 |

MSR AMEVCNTR01\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b001 |

### Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are UNDEFINED.

**Note**

AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR01\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
            HAFGRTR_EL2.AMEVCNTR01_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR0_ELO[1];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVCNTR01_ELO == '1'
            then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);

```



```

else
    X[t, 64] = AMEVCNTR0_EL0[1];
elseif PSTATE.EL == EL2 then
    if CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMEVCNTR0_EL0[1];
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMEVCNTR0_EL0[1];

```

MSR AMEVCNTR01\_EL0, <Xt>

```

if IsHighestEL(PSTATE.EL) then
    AMEVCNTR0_EL0[1] = X[t, 64];
else
    UNDEFINED;

```

### A.11.5 AMEVCNTR02\_EL0, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counter 2.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Activity Monitors registers

##### Access type

See bit descriptions

##### Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

#### Bit descriptions

**Figure A-104: AArch64\_amevcntr02\_el0 bit assignments**

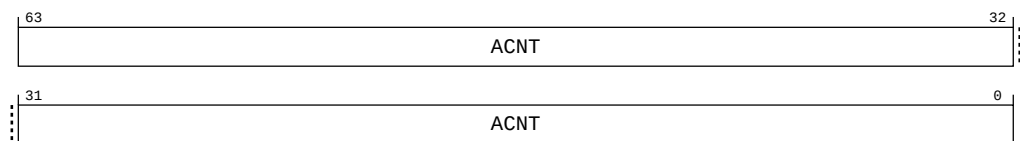


**Table A-293: AMEVCNTR02\_ELO bit descriptions**

| Bits   | Name | Description                                            | Reset              |
|--------|------|--------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of architected activity monitor event counter 2. | 0x0000000000000000 |

### Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR02\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b010 |

MSR AMEVCNTR02\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b010 |

### Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR02\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
            HAFGRTR_EL2.AMEVCNTR02_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);

```

```

        else
            X[t, 64] = AMEVCNTR0_EL0[2];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVCNTR02_EL0 == '1'
            then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                end
            else
                X[t, 64] = AMEVCNTR0_EL0[2];
            elsif PSTATE.EL == EL2 then
                if CPTR_EL3.TAM == '1' then
                    if Halted() && EDSCR.SDD == '1' then
                        UNDEFINED;
                    else
                        AArch64.SystemAccessTrap(EL3, 0x18);
                    end
                else
                    X[t, 64] = AMEVCNTR0_EL0[2];
            elsif PSTATE.EL == EL3 then
                X[t, 64] = AMEVCNTR0_EL0[2];
            end

```

MSR AMEVCNTR02\_EL0, <Xt>

```

if IsHighestEL(PSTATE.EL) then
    AMEVCNTR0_EL0[2] = X[t, 64];
else
    UNDEFINED;
end

```

## A.11.6 AMEVCNTR03\_EL0, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counter 3.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Activity Monitors registers

#### Access type

See bit descriptions

#### Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Bit descriptions

Figure A-105: AArch64\_amevcntr03\_el0 bit assignments

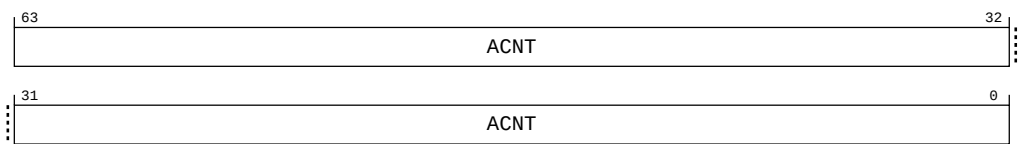


Table A-296: AMEVCNTR03\_ELO bit descriptions

| Bits   | Name | Description                                            | Reset              |
|--------|------|--------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of architected activity monitor event counter 3. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR03\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b011 |

MSR AMEVCNTR03\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0100 | 0b011 |

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_ELO are UNDEFINED.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVCNTR03\_ELO

```
if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
```

```

        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HAFGRTR_EL2.AMEVCNTR03_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR0_ELO[3];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVCNTR03_ELO == '1'
then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR0_ELO[3];
        elsif PSTATE.EL == EL2 then
            if CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR0_ELO[3];
        elsif PSTATE.EL == EL3 then
            X[t, 64] = AMEVCNTR0_ELO[3];

```

MSR AMEVCNTR03\_ELO, <Xt>

```

if IsHighestEL(PSTATE.EL) then
    AMEVCNTR0_ELO[3] = X[t, 64];
else
    UNDEFINED;

```

## A.11.7 AMEVTYPER00\_ELO, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR00\_ELO counts.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

Functional group  
Activity Monitors registers

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0001 | 0001 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-106: AArch64\_amevtyper00\_el0 bit assignments

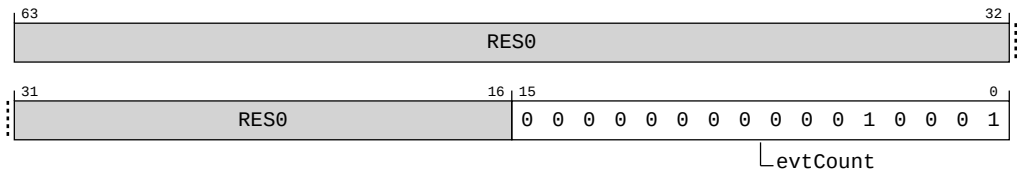


Table A-299: AMEVTYPER00\_ELO bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                               | Reset  |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                  | RES0   |
| [15:0]  | evtCount | Event to count. The event number of the event that is counted by the architected activity monitor event counter AArch64-AMEVCNTR00_ELO. The value of this field is architecturally mandated for each architected counter.<br><br><b>0b00000000000010001</b><br>Processor frequency cycles | 0x0011 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPER0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPER00\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0110 | 0b000 |

## Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPER0<n>\_ELO are UNDEFINED.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPER00\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTYPER0_ELO[0];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    X[t, 64] = AMEVTYPER0_ELO[0];
            elsif PSTATE.EL == EL2 then
                if CPTR_EL3.TAM == '1' then
                    if Halted() && EDSCR.SDD == '1' then
                        UNDEFINED;
                    else
                        AArch64.SystemAccessTrap(EL3, 0x18);
                    else
                        X[t, 64] = AMEVTYPER0_ELO[0];
            elsif PSTATE.EL == EL3 then
                X[t, 64] = AMEVTYPER0_ELO[0];

```

A.11.8 AMEVTYPER01\_ELO, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR01\_ELO counts.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0100 | 0000 | 0000 | 0100 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-107: AArch64\_amevtyper01\_el0 bit assignments

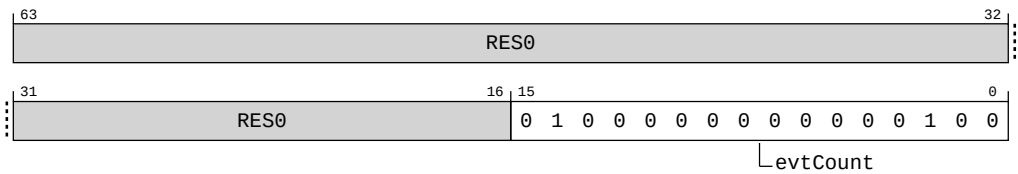


Table A-301: AMEVTYPER01\_ELO bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                       | Reset  |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:16] | RES0     | Reserved                                                                                                                                                                                                                                                                          | RES0   |
| [15:0]  | evtCount | Event to count. The event number of the event that is counted by the architected activity monitor event counter AArch64-AMEVCNTR01_ELO. The value of this field is architecturally mandated for each architected counter.<br><br>0b01000000000000100<br>Constant frequency cycles | 0x4004 |



## Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPEP0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPEP01\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0110 | 0b001 |

## Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPEP0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPEP01\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTYPEP0_ELO[1];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    X[t, 64] = AMEVTYPEP0_ELO[1];
            elsif PSTATE.EL == EL2 then
                if CPTR_EL3.TAM == '1' then
                    if Halted() && EDSCR.SDD == '1' then
                        UNDEFINED;
                    else

```

```
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = AMEVTYPER0_ELO[1];
    elsif PSTATE.EL == EL3 then
        X[t, 64] = AMEVTYPER0_ELO[1];
```

A.11.9 AMEVTYPER02\_ELO, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR02\_ELO counts.

Configurations

This register is available in all configurations.

Attributes

Width

64

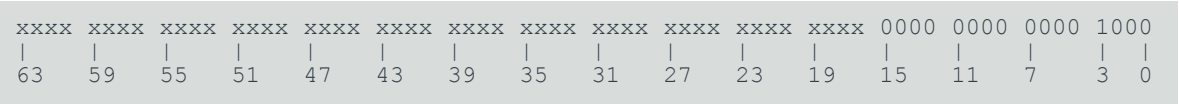
Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-108: AArch64\_amevtyper02\_el0 bit assignments

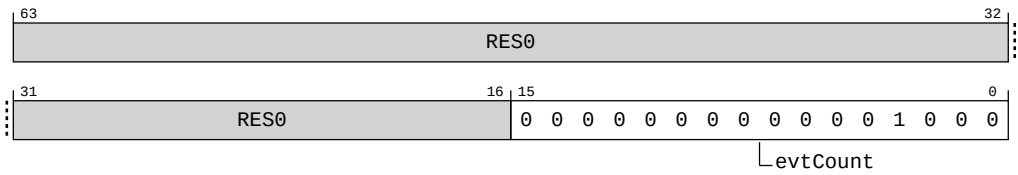


Table A-303: AMEVTYPER02\_ELO bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:16] | RES0 | Reserved    | RES0  |

| Bits   | Name     | Description                                                                                                                                                                                                                                                                         | Reset  |
|--------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:0] | evtCount | Event to count. The event number of the event that is counted by the architected activity monitor event counter AArch64-AMEVCNTR02_ELO. The value of this field is architecturally mandated for each architected counter.<br><br><b>0b00000000000001000</b><br>Instructions retired | 0x0008 |

## Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPER0<n>\_ELO are **UNDEFINED**.



Note

AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPER02\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0110 | 0b010 |

## Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPER0<n>\_ELO are **UNDEFINED**.



Note

AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPER02\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTYPER0_EL0[2];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then

```

```
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = AMEVTYPER0_EL0[2];
elseif PSTATE.EL == EL2 then
    if CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = AMEVTYPER0_EL0[2];
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMEVTYPER0_EL0[2];
```

### A.11.10 AMEVTYPER03\_ELO, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR03\_ELO counts.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Activity Monitors registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0100 | 0000 | 0000 | 0101 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-109: AArch64\_amevtyper03\_el0 bit assignments

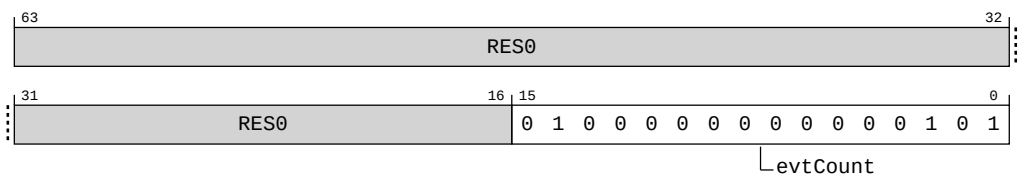


Table A-305: AMEVTYPER03\_ELO bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                 | Reset  |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:16] | RES0     | Reserved                                                                                                                                                                                                                                                                    | RES0   |
| [15:0]  | evtCount | Event to count. The event number of the event that is counted by the architected activity monitor event counter AArch64-AMEVCNTR03_ELO. The value of this field is architecturally mandated for each architected counter.<br><br>0b01000000000000101<br>Memory stall cycles | 0x4005 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPER0<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPER03\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b0110 | 0b011 |

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVTYPER0<n>\_ELO are UNDEFINED.



AArch64-AMCGCR\_ELO.CG0NC identifies the number of architected activity monitor event counters.

MRS <Xt>, AMEVTYPER03\_ELO

```
if PSTATE.EL == EL0 then
```

```

if AMUSERENR_EL0.EN == '0' then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif EL2Enabled() && CPTR_EL2.TAM == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif CPTR_EL3.TAM == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
else
    X[t, 64] = AMEVTYPER0_EL0[3];
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && CPTR_EL2.TAM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
elseif CPTR_EL3.TAM == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
else
    X[t, 64] = AMEVTYPER0_EL0[3];
elseif PSTATE.EL == EL2 then
    if CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = AMEVTYPER0_EL0[3];
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMEVTYPER0_EL0[3];

```

### A.11.11 AMEVCNTR10\_EL0, Activity Monitors Event Counter Registers 1

Provides access to the auxiliary activity monitor event counter 0.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Activity Monitors registers

##### Access type

See bit descriptions

##### Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure A-110: AArch64\_amevcntr10\_el0 bit assignments

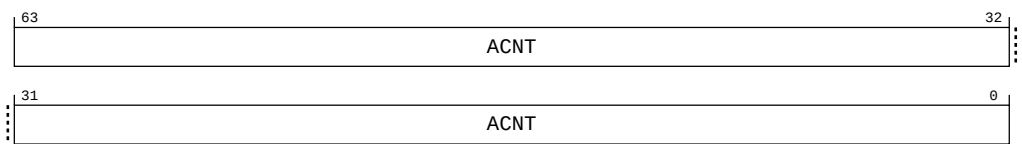


Table A-307: AMEVCNTR10\_ELO bit descriptions

| Bits   | Name | Description                                          | Reset              |
|--------|------|------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of Auxiliary activity monitor event counter 0. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVCNTR1<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVCNTR10\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1100 | 0b000 |

MSR AMEVCNTR10\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1100 | 0b000 |

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVCNTR1<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVCNTR10\_ELO

```
if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
```

```

        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HAFGRTR_EL2.AMEVCNTR10_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[0];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVCNTR10_ELO == '1'
then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[0];
        elsif PSTATE.EL == EL2 then
            if CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[0];
        elsif PSTATE.EL == EL3 then
            X[t, 64] = AMEVCNTR1_ELO[0];

```

MSR AMEVCNTR10\_ELO, <Xt>

```

if IsHighestEL(PSTATE.EL) then
    AMEVCNTR1_ELO[0] = X[t, 64];
else
    UNDEFINED;

```

## A.11.12 AMEVCNTR11\_ELO, Activity Monitors Event Counter Registers 1

Provides access to the auxiliary activity monitor event counter 1.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64



Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure A-111: AArch64\_amevcntr11\_el0 bit assignments

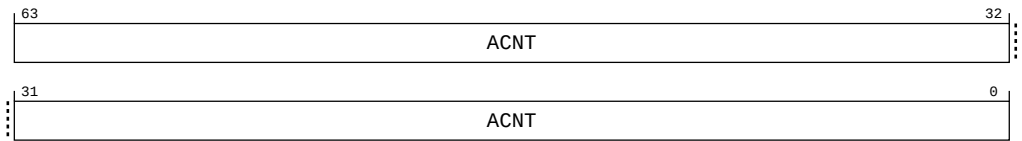


Table A-310: AMEVCNTR11\_ELO bit descriptions

| Bits   | Name | Description                                          | Reset              |
|--------|------|------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of Auxiliary activity monitor event counter 1. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVCNTR1<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVCNTR11\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1100 | 0b001 |

MSR AMEVCNTR11\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1100 | 0b001 |

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVCNTR1<n>\_ELO are UNDEFINED.

**Note**

AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

## MRS <Xt>, AMEVCNTR11\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
            HAFGRTR_EL2.AMEVCNTR11_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[1];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_EL2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVCNTR11_ELO == '1'
            then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    X[t, 64] = AMEVCNTR1_ELO[1];
        elsif PSTATE.EL == EL2 then
            if CPTR_EL3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[1];
        elsif PSTATE.EL == EL3 then
            X[t, 64] = AMEVCNTR1_ELO[1];

```

## MSR AMEVCNTR11\_ELO, <Xt>

```

if IsHighestEL(PSTATE.EL) then
    AMEVCNTR1_ELO[1] = X[t, 64];
else
    UNDEFINED;

```

A.11.13 AMEVCNTR12\_ELO, Activity Monitors Event Counter Registers 1

Provides access to the auxiliary activity monitor event counter 2.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Bit descriptions

Figure A-112: AArch64\_amevcntr12\_el0 bit assignments

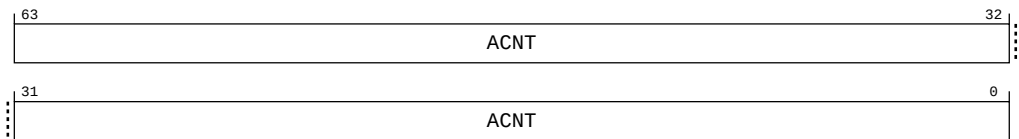


Table A-313: AMEVCNTR12\_ELO bit descriptions

| Bits   | Name | Description                                          | Reset              |
|--------|------|------------------------------------------------------|--------------------|
| [63:0] | ACNT | Value of Auxiliary activity monitor event counter 2. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVCNTR1<n>\_ELO are **UNDEFINED**.



AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVCNTR12\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1100 | 0b010 |

MSR AMEVCNTR12\_ELO, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1100 | 0b010 |

## Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVCNTR1<n>\_ELO are UNDEFINED.



Note

AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVCNTR12\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_ELO2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_ELO2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_ELO2.<E2H,TGE> != '11' && SCR_ELO3.FGTEn == '1' &&
            HAFGRTR_ELO2.AMEVCNTR12_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_ELO3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[2];
        elsif PSTATE.EL == EL1 then
            if EL2Enabled() && CPTR_ELO2.TAM == '1' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif EL2Enabled() && SCR_ELO3.FGTEn == '1' && HAFGRTR_ELO2.AMEVCNTR12_ELO == '1'
            then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif CPTR_ELO3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    X[t, 64] = AMEVCNTR1_ELO[2];
        elsif PSTATE.EL == EL2 then
            if CPTR_ELO3.TAM == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVCNTR1_ELO[2];
        elsif PSTATE.EL == EL3 then

```

```
X[t, 64] = AMEVCNTR1_ELO[2];
```

MSR AMEVCNTR12\_ELO, <Xt>

```
if IsHighestEL(PSTATE.EL) then
    AMEVCNTR1_ELO[2] = X[t, 64];
else
    UNDEFINED;
```

A.11.14 AMEVTYPER10\_ELO, Activity Monitors Event Type Registers 1

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR10\_ELO counts.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0011 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-113: AArch64\_amevtyper10\_el0 bit assignments

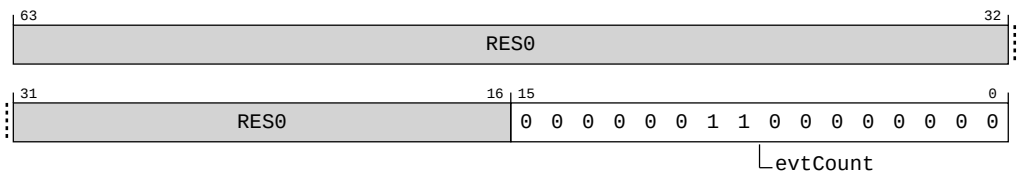


**Table A-316: AMEVTYPER10\_ELO bit descriptions**

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset  |
|---------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RES0   |
| [15:0]  | evtCount | <p>Event to count. The event number of the event that is counted by the auxiliary activity monitor event counter AArch64-AMEVCNTR1&lt;n&gt;_ELO.</p> <p>It is IMPLEMENTATION DEFINED what values are supported by each counter.</p> <p>If software writes a value to this field which is not supported by the corresponding counter AArch64-AMEVCNTR1&lt;n&gt;_ELO, then:</p> <ul style="list-style-type: none"> <li>It is UNPREDICTABLE which event will be counted.</li> <li>The value read back is <b>UNKNOWN</b>.</li> </ul> <p>The event counted by AArch64-AMEVCNTR1&lt;n&gt;_ELO might be fixed at implementation. In this case, the field is read-only and writes are UNDEFINED.</p> <p>If the corresponding counter AArch64-AMEVCNTR1&lt;n&gt;_ELO is enabled, writes to this register have UNPREDICTABLE results.</p> <p><b>0b00000001100000000</b><br/>Gear 0 (MPMM bank 0) period threshold exceeded</p> | 0x0300 |

### Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVTYPER1<n>\_ELO are **UNDEFINED**.

**Note**

AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVTYPER10\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1110 | 0b000 |

### Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVTYPER1<n>\_ELO are UNDEFINED.

**Note**

AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

## MRS &lt;Xt&gt;, AMEVTYPER10\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HAFGRTR_EL2.AMEVTYPER10_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTYPER1_ELO[0];
    elsif PSTATE.EL == EL1 then
        if EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVTYPER10_ELO == '1'
then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTYPER1_ELO[0];
    elsif PSTATE.EL == EL2 then
        if CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTYPER1_ELO[0];
    elsif PSTATE.EL == EL3 then
        X[t, 64] = AMEVTYPER1_ELO[0];

```

### A.11.15 AMEVTYPER11\_ELO, Activity Monitors Event Type Registers 1

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR11\_ELO counts.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0011 | 0000 | 0001 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-114: AArch64\_amevtyper11\_el0 bit assignments

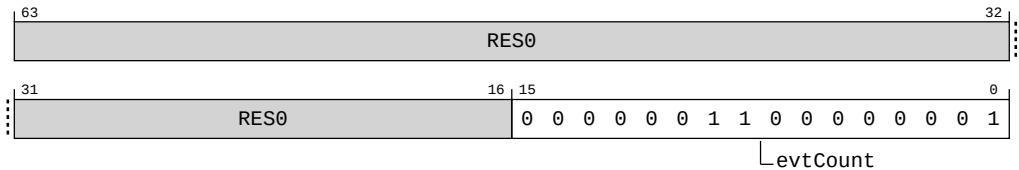


Table A-318: AMEVTYPER11\_EL0 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset  |
|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RES0   |
| [15:0]  | evtCount | <p>Event to count. The event number of the event that is counted by the auxiliary activity monitor event counter AArch64-AMEVCNTR1&lt;n&gt;_ELO.</p> <p>It is IMPLEMENTATION DEFINED what values are supported by each counter.</p> <p>If software writes a value to this field which is not supported by the corresponding counter AArch64-AMEVCNTR1&lt;n&gt;_ELO, then:</p> <ul style="list-style-type: none"><li>It is UNPREDICTABLE which event will be counted.</li><li>The value read back is <b>UNKNOWN</b>.</li></ul> <p>The event counted by AArch64-AMEVCNTR1&lt;n&gt;_ELO might be fixed at implementation. In this case, the field is read-only and writes are UNDEFINED.</p> <p>If the corresponding counter AArch64-AMEVCNTR1&lt;n&gt;_ELO is enabled, writes to this register have UNPREDICTABLE results.</p> <p><b>0b0000000110000001</b></p> <p>Gear 1 (MPMM bank 1) period threshold exceeded</p> | 0x0301 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVTYPER1<n>\_ELO are **UNDEFINED**.





AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

## MRS &lt;Xt&gt;, AMEVTYPER11\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1110 | 0b001 |

## Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVTYPER1<n>\_ELO are UNDEFINED.



AArch64-AMCGCR\_EL0.CG1NC identifies the number of auxiliary activity monitor event counters.

## MRS &lt;Xt&gt;, AMEVTYPER11\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HAFGRTR_EL2.AMEVTPER11_ELO == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTPER1_ELO[1];
    elsif PSTATE.EL == EL1 then
        if EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVTPER11_ELO == '1'
then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMEVTPER1_ELO[1];
    elsif PSTATE.EL == EL2 then
        if CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);

```

```
else
    X[t, 64] = AMEVTYPER1_EL0[1];
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMEVTYPER1_EL0[1];
```

A.11.16 AMEVTYPER12\_EL0, Activity Monitors Event Type Registers 1

Provides information on the events that an architected activity monitor event counter AArch64-AMEVCNTR12\_EL0 counts.

Configurations

This register is available in all configurations.

Attributes

Width

64

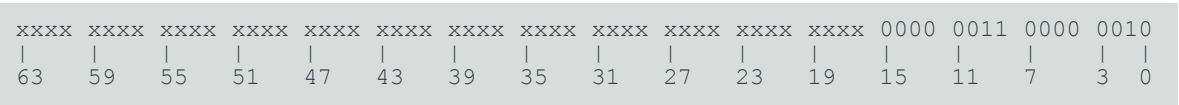
Functional group

Activity Monitors registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-115: AArch64\_amevtyper12\_el0 bit assignments

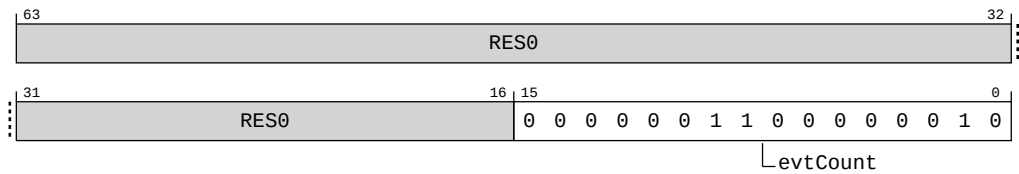


Table A-320: AMEVTYPER12\_EL0 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:16] | RES0 | Reserved    | RES0  |

| Bits   | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset  |
|--------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:0] | evtCount | <p>Event to count. The event number of the event that is counted by the auxiliary activity monitor event counter AArch64-AMEVCNTR1&lt;n&gt;_ELO.</p> <p>It is IMPLEMENTATION DEFINED what values are supported by each counter.</p> <p>If software writes a value to this field which is not supported by the corresponding counter AArch64-AMEVCNTR1&lt;n&gt;_ELO, then:</p> <ul style="list-style-type: none"> <li>It is UNPREDICTABLE which event will be counted.</li> <li>The value read back is <b>UNKNOWN</b>.</li> </ul> <p>The event counted by AArch64-AMEVCNTR1&lt;n&gt;_ELO might be fixed at implementation. In this case, the field is read-only and writes are UNDEFINED.</p> <p>If the corresponding counter AArch64-AMEVCNTR1&lt;n&gt;_ELO is enabled, writes to this register have UNPREDICTABLE results.</p> <p><b>0b00000001100000010</b><br/>Gear 2 (MPMM bank 2) period threshold exceeded</p> | 0x0302 |

## Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVTYPER1<n>\_ELO are **UNDEFINED**.



Note

AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVTYPER12\_ELO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b011 | 0b1101 | 0b1110 | 0b010 |

## Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads and writes of AMEVTYPER1<n>\_ELO are UNDEFINED.



Note

AArch64-AMCGCR\_ELO.CG1NC identifies the number of auxiliary activity monitor event counters.

MRS <Xt>, AMEVTYPER12\_ELO

```

if PSTATE.EL == EL0 then
    if AMUSERENR_ELO.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);

```

```

else
    AArch64.SystemAccessTrap(EL1, 0x18);
elseif EL2Enabled() && CPTR_EL2.TAM == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' && SCR_EL3.FGTEn == '1' &&
HAFGRTR_EL2.AMEVTPER12_EL0 == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif CPTR_EL3.TAM == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = AMEVTPER1_EL0[2];
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && CPTR_EL2.TAM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HAFGRTR_EL2.AMEVTPER12_EL0 == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMEVTPER1_EL0[2];
elseif PSTATE.EL == EL2 then
    if CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMEVTPER1_EL0[2];
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMEVTPER1_EL0[2];

```

## A.12 AArch64 Trace unit registers summary

The following summary table provides an overview of all Trace unit registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-322: Trace unit registers summary**

| Name         | Op0 | Op1 | CRn | CRm      | Op2 | Reset                      | Width  | Description          |
|--------------|-----|-----|-----|----------|-----|----------------------------|--------|----------------------|
| TRCIMSPEC_n_ | 2   | 1   | C0  | 0:m[2:0] | 7   | See individual bit resets. | 64-bit | IMP DEF Register <n> |
| TRCTRAIDR    | 2   | 1   | C0  | C0       | 1   | See individual bit resets. | 64-bit | Trace ID Register    |

| Name         | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                     |
|--------------|-----|-----|-----|-----|-----|----------------------------|--------|-------------------------------------------------|
| TRCVICTLR    | 2   | 1   | C0  | C0  | 2   | See individual bit resets. | 64-bit | ViewInst Main Control Register                  |
| TRCSEQEVRO   | 2   | 1   | C0  | C0  | 4   | See individual bit resets. | 64-bit | Sequencer State Transition Control Register <n> |
| TRCCNTRLDVRO | 2   | 1   | C0  | C0  | 5   | See individual bit resets. | 64-bit | Counter Reload Value Register <n>               |
| TRCIDR8      | 2   | 1   | C0  | C0  | 6   | See individual bit resets. | 64-bit | ID Register 8                                   |
| TRCIMSPEC0   | 2   | 1   | C0  | C0  | 7   | See individual bit resets. | 64-bit | IMP DEF Register 0                              |
| TRCPRGCTLR   | 2   | 1   | C0  | C1  | 0   | See individual bit resets. | 64-bit | Programming Control Register                    |
| TRCVIIECTLR  | 2   | 1   | C0  | C1  | 2   | See individual bit resets. | 64-bit | ViewInst Include/Exclude Control Register       |
| TRCSEQEVR1   | 2   | 1   | C0  | C1  | 4   | See individual bit resets. | 64-bit | Sequencer State Transition Control Register <n> |
| TRCCNTRLDVR1 | 2   | 1   | C0  | C1  | 5   | See individual bit resets. | 64-bit | Counter Reload Value Register <n>               |
| TRCIDR9      | 2   | 1   | C0  | C1  | 6   | See individual bit resets. | 64-bit | ID Register 9                                   |
| TRCVISSCTLR  | 2   | 1   | C0  | C2  | 2   | See individual bit resets. | 64-bit | ViewInst Start/Stop Control Register            |
| TRCSEQEVR2   | 2   | 1   | C0  | C2  | 4   | See individual bit resets. | 64-bit | Sequencer State Transition Control Register <n> |
| TRCIDR10     | 2   | 1   | C0  | C2  | 6   | See individual bit resets. | 64-bit | ID Register 10                                  |
| TRCSTATR     | 2   | 1   | C0  | C3  | 0   | See individual bit resets. | 64-bit | Trace Status Register                           |
| TRCIDR11     | 2   | 1   | C0  | C3  | 6   | See individual bit resets. | 64-bit | ID Register 11                                  |
| TRCCONFIGR   | 2   | 1   | C0  | C4  | 0   | See individual bit resets. | 64-bit | Trace Configuration Register                    |
| TRCCNTCTLRO  | 2   | 1   | C0  | C4  | 5   | See individual bit resets. | 64-bit | Counter Control Register <n>                    |
| TRCIDR12     | 2   | 1   | C0  | C4  | 6   | See individual bit resets. | 64-bit | ID Register 12                                  |
| TRCCNTCTLR1  | 2   | 1   | C0  | C5  | 5   | See individual bit resets. | 64-bit | Counter Control Register <n>                    |
| TRCIDR13     | 2   | 1   | C0  | C5  | 6   | See individual bit resets. | 64-bit | ID Register 13                                  |
| TRCAUXCTLR   | 2   | 1   | C0  | C6  | 0   | See individual bit resets. | 64-bit | Auxiliary Control Register                      |
| TRCSEQRSTEV  | 2   | 1   | C0  | C6  | 4   | See individual bit resets. | 64-bit | Sequencer Reset Control Register                |
| TRCSEQSTR    | 2   | 1   | C0  | C7  | 4   | See individual bit resets. | 64-bit | Sequencer State Register                        |

| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                 |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|---------------------------------------------|
| TRCEVENTCTLOR | 2   | 1   | C0  | C8  | 0   | See individual bit resets. | 64-bit | Event Control 0 Register                    |
| TRCEXTINSELRO | 2   | 1   | C0  | C8  | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCCNTVR0     | 2   | 1   | C0  | C8  | 5   | See individual bit resets. | 64-bit | Counter Value Register <n>                  |
| TRCIDR0       | 2   | 1   | C0  | C8  | 7   | See individual bit resets. | 64-bit | ID Register 0                               |
| TRCEVENTCTL1R | 2   | 1   | C0  | C9  | 0   | See individual bit resets. | 64-bit | Event Control 1 Register                    |
| TRCEXTINSELR1 | 2   | 1   | C0  | C9  | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCCNTVR1     | 2   | 1   | C0  | C9  | 5   | See individual bit resets. | 64-bit | Counter Value Register <n>                  |
| TRCIDR1       | 2   | 1   | C0  | C9  | 7   | See individual bit resets. | 64-bit | ID Register 1                               |
| TRCRSR        | 2   | 1   | C0  | C10 | 0   | See individual bit resets. | 64-bit | Resources Status Register                   |
| TRCEXTINSELR2 | 2   | 1   | C0  | C10 | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCIDR2       | 2   | 1   | C0  | C10 | 7   | See individual bit resets. | 64-bit | ID Register 2                               |
| TRCEXTINSELR3 | 2   | 1   | C0  | C11 | 4   | See individual bit resets. | 64-bit | External Input Select Register <n>          |
| TRCIDR3       | 2   | 1   | C0  | C11 | 7   | See individual bit resets. | 64-bit | ID Register 3                               |
| TRCTSCTLR     | 2   | 1   | C0  | C12 | 0   | See individual bit resets. | 64-bit | Timestamp Control Register                  |
| TRCIDR4       | 2   | 1   | C0  | C12 | 7   | See individual bit resets. | 64-bit | ID Register 4                               |
| TRCSYNCPR     | 2   | 1   | C0  | C13 | 0   | See individual bit resets. | 64-bit | Synchronization Period Register             |
| TRCIDR5       | 2   | 1   | C0  | C13 | 7   | See individual bit resets. | 64-bit | ID Register 5                               |
| TRCCCCTLR     | 2   | 1   | C0  | C14 | 0   | See individual bit resets. | 64-bit | Cycle Count Control Register                |
| TRCIDR6       | 2   | 1   | C0  | C14 | 7   | See individual bit resets. | 64-bit | ID Register 6                               |
| TRCBBCTLR     | 2   | 1   | C0  | C15 | 0   | See individual bit resets. | 64-bit | Branch Broadcast Control Register           |
| TRCIDR7       | 2   | 1   | C0  | C15 | 7   | See individual bit resets. | 64-bit | ID Register 7                               |
| TRCSSCCRO     | 2   | 1   | C1  | C0  | 2   | See individual bit resets. | 64-bit | Single-shot Comparator Control Register <n> |
| TRCOSLSR      | 2   | 1   | C1  | C1  | 4   | See individual bit resets. | 64-bit | Trace OS Lock Status Register               |

| Name        | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                        |
|-------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------|
| TRCRSCTLR2  | 2   | 1   | C1  | C2  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR3  | 2   | 1   | C1  | C3  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR4  | 2   | 1   | C1  | C4  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR5  | 2   | 1   | C1  | C5  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR6  | 2   | 1   | C1  | C6  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR7  | 2   | 1   | C1  | C7  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR8  | 2   | 1   | C1  | C8  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCSSCSR0   | 2   | 1   | C1  | C8  | 2   | See individual bit resets. | 64-bit | Single-shot Comparator Control Status Register <n> |
| TRCRSCTLR9  | 2   | 1   | C1  | C9  | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR10 | 2   | 1   | C1  | C10 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR11 | 2   | 1   | C1  | C11 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR12 | 2   | 1   | C1  | C12 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR13 | 2   | 1   | C1  | C13 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR14 | 2   | 1   | C1  | C14 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCRSCTLR15 | 2   | 1   | C1  | C15 | 0   | See individual bit resets. | 64-bit | Resource Selection Control Register <n>            |
| TRCACVR0    | 2   | 1   | C2  | C0  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR0   | 2   | 1   | C2  | C0  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR1    | 2   | 1   | C2  | C2  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR1   | 2   | 1   | C2  | C2  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR2    | 2   | 1   | C2  | C4  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR2   | 2   | 1   | C2  | C4  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |
| TRCACVR3    | 2   | 1   | C2  | C6  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>              |
| TRCACATR3   | 2   | 1   | C2  | C6  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>        |

| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                              |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------------------|
| TRCACVR4      | 2   | 1   | C2  | C8  | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR4     | 2   | 1   | C2  | C8  | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCACVR5      | 2   | 1   | C2  | C10 | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR5     | 2   | 1   | C2  | C10 | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCACVR6      | 2   | 1   | C2  | C12 | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR6     | 2   | 1   | C2  | C12 | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCACVR7      | 2   | 1   | C2  | C14 | 0   | See individual bit resets. | 64-bit | Address Comparator Value Register <n>                    |
| TRCACATR7     | 2   | 1   | C2  | C14 | 2   | See individual bit resets. | 64-bit | Address Comparator Access Type Register <n>              |
| TRCCIDCVR0    | 2   | 1   | C3  | C0  | 0   | See individual bit resets. | 64-bit | Context Identifier Comparator Value Registers <n>        |
| TRCVMIDCVR0   | 2   | 1   | C3  | C0  | 1   | See individual bit resets. | 64-bit | Virtual Context Identifier Comparator Value Register <n> |
| TRCCIDCCTLR0  | 2   | 1   | C3  | C0  | 2   | See individual bit resets. | 64-bit | Context Identifier Comparator Control Register 0         |
| TRCVMIDCCTLR0 | 2   | 1   | C3  | C2  | 2   | See individual bit resets. | 64-bit | Virtual Context Identifier Comparator Control Register 0 |
| TRCDEVID      | 2   | 1   | C7  | C2  | 7   | See individual bit resets. | 64-bit | Device Configuration Register                            |
| TRCCLAIMSET   | 2   | 1   | C7  | C8  | 6   | See individual bit resets. | 64-bit | Claim Tag Set Register                                   |
| TRCCLAIMCLR   | 2   | 1   | C7  | C9  | 6   | See individual bit resets. | 64-bit | Claim Tag Clear Register                                 |
| TRCAUTHSTATUS | 2   | 1   | C7  | C14 | 6   | See individual bit resets. | 64-bit | Authentication Status Register                           |
| TRCDEVARCH    | 2   | 1   | C7  | C15 | 6   | See individual bit resets. | 64-bit | Device Architecture Register                             |

## A.12.1 TRCIDR8, ID Register 8

Returns the maximum speculation depth of the instruction trace element stream.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64



**Functional group**  
Trace unit registers

**Access type**  
See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

**Bit descriptions**

**Figure A-116: AArch64\_trcidr8 bit assignments**

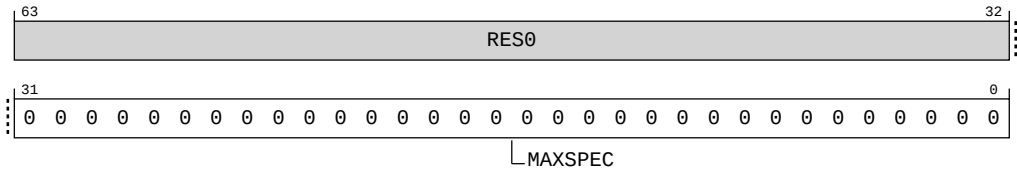


Table A-323: TRCIDR8 bit descriptions

| Bits    | Name    | Description                                                                                                                                                                                                                                                                         | Reset      |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [63:32] | RES0    | Reserved                                                                                                                                                                                                                                                                            | RES0       |
| [31:0]  | MAXSPEC | Indicates the maximum speculation depth of the instruction trace element stream. This is the maximum number of PO elements in the trace element stream that can be speculative at any time.<br><br>0b00000000000000000000000000000000<br>No speculation in the trace element stream | 0x00000000 |

**Access**  
MRS <Xt>, TRCIDR8

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0000 | 0b110 |

**Accessibility**  
MRS <Xt>, TRCIDR8

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
```

```
    AArch64.SystemAccessTrap(EL1, 0x18);
elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif CPTR_EL3.TTA == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR8;
elseif PSTATE.EL == EL2 then
    if CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR8;
elseif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR8;
```

### A.12.2 TRCIMSPECO, IMP DEF Register 0

TRCIMSPECO shows the presence of any **IMPLEMENTATION DEFINED** features, and provides an interface to enable the features that are provided.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Trace unit registers

##### Access type

RO

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-117: AArch64\_trcimspec0 bit assignments

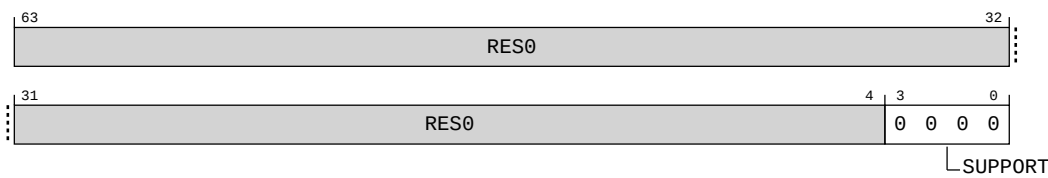


Table A-325: TRCIMSPEC0 bit descriptions

| Bits   | Name    | Description                                                                                                                                                            | Reset  |
|--------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:4] | RES0    | Reserved                                                                                                                                                               | RES0   |
| [3:0]  | SUPPORT | Indicates whether the implementation supports <b>IMPLEMENTATION DEFINED</b> features.<br><br><b>0b0000</b><br>No <b>IMPLEMENTATION DEFINED</b> features are supported. | 0b0000 |

Access

MRS <Xt>, TRCIMSPEC0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0000 | 0b111 |

MSR TRCIMSPEC0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0000 | 0b111 |

Accessibility

MRS <Xt>, TRCIMSPEC0

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCIMSPEcn == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIMSPEC0;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
```

```

        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIMSPEC0;
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIMSPEC0;

```

MSR TRCIMSPEC0, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDBGWTR_EL2.TRCIMSPEcn == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCIMSPEC0 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCIMSPEC0 = X[t, 64];
elseif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCIMSPEC0 = X[t, 64];

```

### A.12.3 TRCIDR10, ID Register 10

Returns the tracing capabilities of the trace unit.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Trace unit registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-118: AArch64\_trcidr10 bit assignments

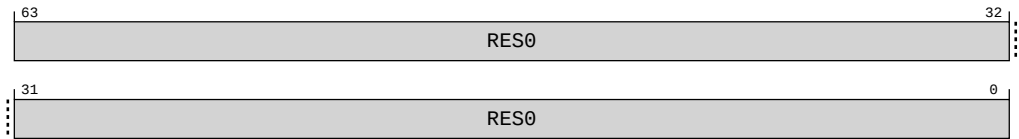


Table A-328: TRCIDR10 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, TRCIDR10

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0010 | 0b110 |

Accessibility

MRS <Xt>, TRCIDR10

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        end
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    end
else
    AArch64.SystemAccessTrap(EL3, 0x18);
end
```

```
        X[t, 64] = TRCIDR10;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDR10;
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR10;
```

### A.12.4 TRCIDR11, ID Register 11

Returns the tracing capabilities of the trace unit.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Trace unit registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-119: AArch64\_trcidr11 bit assignments

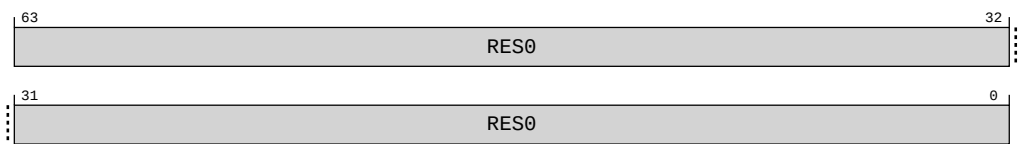


Table A-330: TRCIDR11 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, TRCIDR11

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0011 | 0b110 |

Accessibility

MRS <Xt>, TRCIDR11

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR11;
    elsif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDR11;
    elsif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR11;
```

A.12.5 TRCIDR12, ID Register 12

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

64

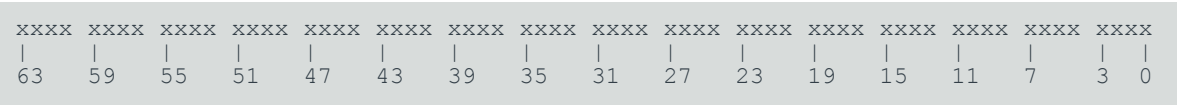
Functional group

Trace unit registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-120: AArch64\_trcidr12 bit assignments

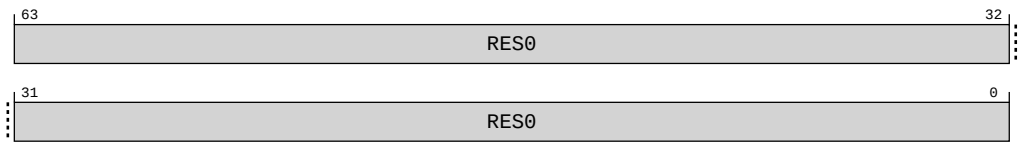


Table A-332: TRCIDR12 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, TRCIDR12

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0100 | 0b110 |



## Accessibility

MRS <Xt>, TRCIDR12

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR12;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDR12;
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR12;

```

## A.12.6 TRCIDR13, ID Register 13

Returns the tracing capabilities of the trace unit.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

Trace unit registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-121: AArch64\_trcidr13 bit assignments

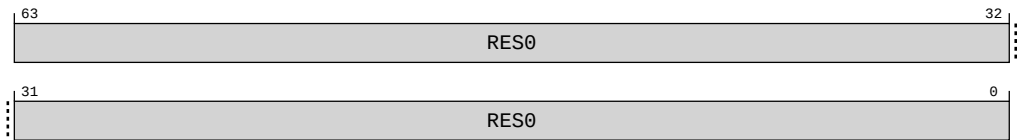


Table A-334: TRCIDR13 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

MRS <Xt>, TRCIDR13

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0101 | 0b110 |

Accessibility

MRS <Xt>, TRCIDR13

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR13;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR13;
```

```
elseif PSTATE.EL == EL3 then
  if CPTR_EL3.TTA == '1' then
    AArch64.SystemAccessTrap(EL3, 0x18);
  else
    X[t, 64] = TRCIDR13;
```

A.12.7 TRCIDR0, ID Register 0

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Trace unit registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | x010 | 1000 | 1xxx | xxx0 | 00xx | 111x | 1010 | 000x |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-122: AArch64\_trcidr0 bit assignments

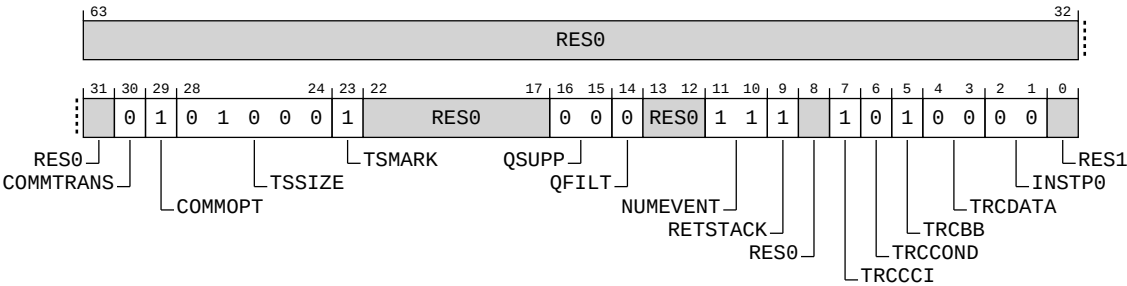


**Table A-336: TRCIDR0 bit descriptions**

| Bits    | Name      | Description                                                                                                                                                                                                                                                   | Reset   |
|---------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [63:31] | RES0      | Reserved                                                                                                                                                                                                                                                      | RES0    |
| [30]    | COMMTRANS | Transaction Start element behavior.<br><b>0b0</b><br>Transaction Start elements are P0 elements.                                                                                                                                                              | 0b0     |
| [29]    | COMMOPT   | Indicates the contents and encodings of Cycle count packets.<br><b>0b1</b><br>Commit mode 1.<br><br>Access to this field is: <b>RAO/WI</b>                                                                                                                    | 0b1     |
| [28:24] | TSSIZE    | Indicates that the trace unit implements Global timestamping and the size of the timestamp value.<br><b>0b01000</b><br>Global timestamping implemented with a 64-bit timestamp value.                                                                         | 0b01000 |
| [23]    | TSMARK    | Indicates whether Timestamp Marker elements are generated.<br><b>0b1</b><br>Timestamp Marker elements are generated.                                                                                                                                          | 0b1     |
| [22:17] | RES0      | Reserved                                                                                                                                                                                                                                                      | RES0    |
| [16:15] | QSUPP     | Indicates that the trace unit implements Q element support.<br><b>0b00</b><br>Q element support is not implemented.                                                                                                                                           | 0b00    |
| [14]    | QFILT     | Indicates if the trace unit implements Q element filtering.<br><b>0b0</b><br>Q element filtering is not implemented.                                                                                                                                          | 0b0     |
| [13:12] | RES0      | Reserved                                                                                                                                                                                                                                                      | RES0    |
| [11:10] | NUMEVENT  | Indicates the number of ETEEvents implemented.<br><b>0b11</b><br>The trace unit supports 4 ETEEvents.                                                                                                                                                         | 0b11    |
| [9]     | RETSTACK  | Indicates if the trace unit supports the return stack.<br><b>0b1</b><br>Return stack implemented.                                                                                                                                                             | 0b1     |
| [8]     | RES0      | Reserved                                                                                                                                                                                                                                                      | RES0    |
| [7]     | TRCCCI    | Indicates if the trace unit implements cycle counting.<br><b>0b1</b><br>Cycle counting implemented.                                                                                                                                                           | 0b1     |
| [6]     | TRCCOND   | Indicates if the trace unit implements conditional instruction tracing. Conditional instruction tracing is not implemented in ETE and this field is reserved for other trace architectures.<br><b>0b0</b><br>Conditional instruction tracing not implemented. | 0b0     |
| [5]     | TRCBB     | Indicates if the trace unit implements branch broadcasting.<br><b>0b1</b><br>Branch broadcasting implemented.                                                                                                                                                 | 0b1     |

| Bits  | Name    | Description                                                                                                                                                                                                                                                                 | Reset |
|-------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [4:3] | TRCDATA | Indicates if the trace unit implements data tracing. Data tracing is not implemented in ETE and this field is reserved for other trace architectures.<br><br><b>0b00</b><br>Tracing of data addresses and data values is not implemented.                                   | 0b00  |
| [2:1] | INSTPO  | Indicates if load and store instructions are PO instructions. Load and store instructions as PO instructions is not implemented in ETE and this field is reserved for other trace architectures.<br><br><b>0b00</b><br>Load and store instructions are not PO instructions. | 0b00  |
| [0]   | RES1    | Reserved                                                                                                                                                                                                                                                                    | RES1  |

## Access

MRS <Xt>, TRCIDRO

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b1000 | 0b111 |

## Accessibility

MRS <Xt>, TRCIDRO

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDRO;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDRO;
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDRO;

```

A.12.8 TRCIDR1, ID Register 1

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Trace unit registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0100 | 0001 | xxxx | xxxx | xxxx | 1111 | 1111 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-123: AArch64\_trcidr1 bit assignments

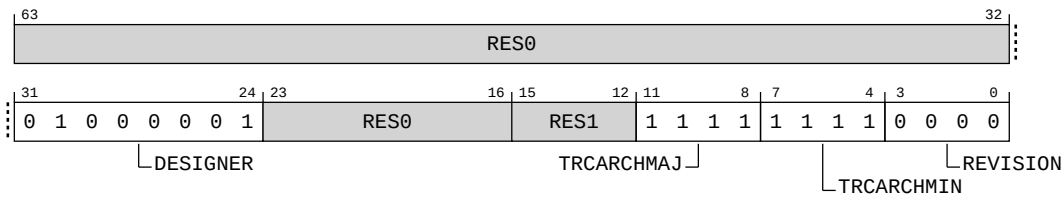


Table A-338: TRCIDR1 bit descriptions

| Bits    | Name     | Description                                                                                                                                                        | Reset |
|---------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0     | Reserved                                                                                                                                                           | RES0  |
| [31:24] | DESIGNER | Indicates which company designed the trace unit. The permitted values of this field are the same as AArch64-MIDR_EL1.Implementer.<br><br>0b01000001<br>Arm Limited | 0x41  |
| [23:16] | RES0     | Reserved                                                                                                                                                           | RES0  |
| [15:12] | RES1     | Reserved                                                                                                                                                           | RES1  |

| Bits   | Name       | Description                                                                                                                    | Reset  |
|--------|------------|--------------------------------------------------------------------------------------------------------------------------------|--------|
| [11:8] | TRCARCHMAJ | Major architecture version.<br><br><b>0b1111</b><br>If both TRCARCHMAJ and TRCARCHMIN == 0xF then refer to AArch64-TRCDEVARCH. | 0b1111 |
| [7:4]  | TRCARCHMIN | Minor architecture version.<br><br><b>0b1111</b><br>If both TRCARCHMAJ and TRCARCHMIN == 0xF then refer to AArch64-TRCDEVARCH. | 0b1111 |
| [3:0]  | REVISION   | Implementation revision that identifies the revision of the trace and OS Lock registers.<br><br><b>0b0000</b><br>Revision 0    | 0b0000 |

### Access

MRS <Xt>, TRCIDR1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b1001 | 0b111 |

### Accessibility

MRS <Xt>, TRCIDR1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        end
    else
        X[t, 64] = TRCIDR1;
    end
elsif PSTATE.EL == EL2 then
    if CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        end
    else
        X[t, 64] = TRCIDR1;
    end
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR1;
    end
end

```

A.12.9 TRCIDR2, ID Register 2

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Trace unit registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1100 | 000x | xxxx | xxxx | x001 | 0000 | 1000 | 1000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-124: AArch64\_trcidr2 bit assignments

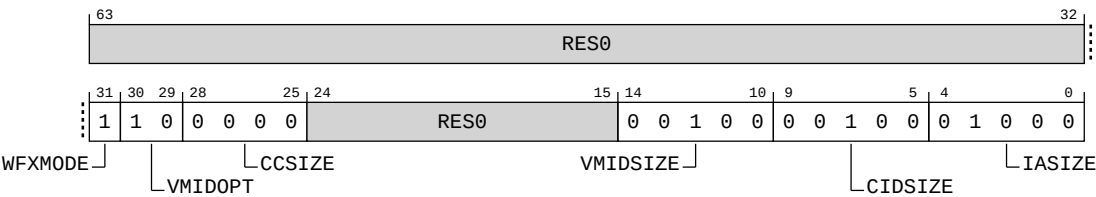


Table A-340: TRCIDR2 bit descriptions

| Bits    | Name    | Description                                                                                                                                                                           | Reset |
|---------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0    | Reserved                                                                                                                                                                              | RES0  |
| [31]    | WFXMODE | Indicates whether WFI, WFIT, WFE, and WFET instructions are classified as P0 instructions:<br><br>0b1<br><br>WFI, WFIT, WFE, and WFET instructions are classified as P0 instructions. | 0b1   |



| Bits    | Name     | Description                                                                                                                                                                           | Reset   |
|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [30:29] | VMIDOPT  | Indicates the options for Virtual context identifier selection.<br><br><b>0b10</b><br>Virtual context identifier selection not supported. AArch64-TRCCONFIGR.VMIDOPT is <b>RES1</b> . | 0b10    |
| [28:25] | CCSIZE   | Indicates the size of the cycle counter.<br><br><b>0b0000</b><br>The cycle counter is 12 bits in length.                                                                              | 0b0000  |
| [24:15] | RES0     | Reserved                                                                                                                                                                              | RES0    |
| [14:10] | VMIDSIZE | Indicates the trace unit Virtual context identifier size.<br><br><b>0b00100</b><br>32-bit Virtual context identifier size.                                                            | 0b00100 |
| [9:5]   | CIDSIZE  | Indicates the Context identifier size.<br><br><b>0b00100</b><br>32-bit Context identifier size.                                                                                       | 0b00100 |
| [4:0]   | IASIZE   | Virtual instruction address size.<br><br><b>0b01000</b><br>Maximum of 64-bit instruction address size.                                                                                | 0b01000 |

## Access

MRS <Xt>, TRCIDR2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b1010 | 0b111 |

## Accessibility

MRS <Xt>, TRCIDR2

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR2;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDR2;

```

```
elseif PSTATE.EL == EL3 then
  if CPTR_EL3.TTA == '1' then
    AArch64.SystemAccessTrap(EL3, 0x18);
  else
    X[t, 64] = TRCIDR2;
```

A.12.10 TRCIDR3, ID Register 3

Returns the base architecture of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Trace unit registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0001 | x111 | 1111 | xx00 | 0000 | 0000 | 0100 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-125: AArch64\_trcidr3 bit assignments

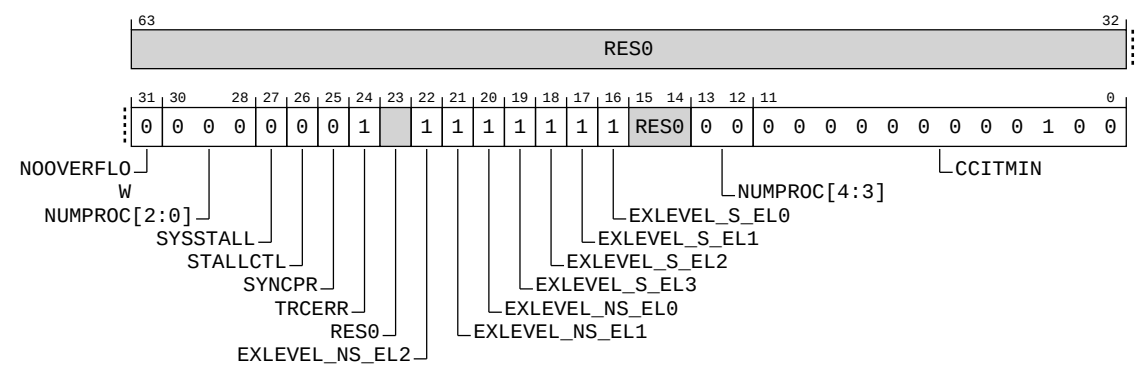


**Table A-342: TRCIDR3 bit descriptions**

| Bits    | Name           | Description                                                                                                                                                            | Reset       |
|---------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [63:32] | <b>RES0</b>    | Reserved                                                                                                                                                               | <b>RES0</b> |
| [31]    | NOOVERFLOW     | Indicates if overflow prevention is implemented.<br><b>0b0</b><br>Overflow prevention is not implemented.                                                              | 0b0         |
| [27]    | SYSSTALL       | Indicates if stalling of the PE is permitted.<br><b>0b0</b><br>Stalling of the PE is not permitted.                                                                    | 0b0         |
| [26]    | STALLCTL       | Indicates if trace unit implements stalling of the PE.<br><b>0b0</b><br>Stalling of the PE is not implemented.                                                         | 0b0         |
| [25]    | SYNCPR         | Indicates if an implementation has a fixed synchronization period.<br><b>0b0</b><br>AArch64-TRCSYNCPR is read/write so software can change the synchronization period. | 0b0         |
| [24]    | TRCERR         | Indicates forced tracing of System Error exceptions is implemented.<br><b>0b1</b><br>Forced tracing of System Error exceptions is implemented.                         | 0b1         |
| [23]    | <b>RES0</b>    | Reserved                                                                                                                                                               | <b>RES0</b> |
| [22]    | EXLEVEL_NS_EL2 | Indicates if Non-secure EL2 is implemented.<br><b>0b1</b><br>Non-secure EL2 is implemented.                                                                            | 0b1         |
| [21]    | EXLEVEL_NS_EL1 | Indicates if Non-secure EL1 is implemented.<br><b>0b1</b><br>Non-secure EL1 is implemented.                                                                            | 0b1         |
| [20]    | EXLEVEL_NS_ELO | Indicates if Non-secure ELO is implemented.<br><b>0b1</b><br>Non-secure ELO is implemented.                                                                            | 0b1         |
| [19]    | EXLEVEL_S_EL3  | Indicates if EL3 is implemented.<br><b>0b1</b><br>EL3 is implemented.                                                                                                  | 0b1         |
| [18]    | EXLEVEL_S_EL2  | Indicates if Secure EL2 is implemented.<br><b>0b1</b><br>Secure EL2 is implemented.                                                                                    | 0b1         |
| [17]    | EXLEVEL_S_EL1  | Indicates if Secure EL1 is implemented.<br><b>0b1</b><br>Secure EL1 is implemented.                                                                                    | 0b1         |
| [16]    | EXLEVEL_S_ELO  | Indicates if Secure ELO is implemented.<br><b>0b1</b><br>Secure ELO is implemented.                                                                                    | 0b1         |
| [15:14] | <b>RES0</b>    | Reserved                                                                                                                                                               | <b>RES0</b> |

| Bits           | Name    | Description                                                                                                                                                                                                                                                          | Reset    |
|----------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [13:12, 30:28] | NUMPROC | Indicates the number of PEs available for tracing.<br><br><b>0b000000</b><br>The trace unit can trace one PE.                                                                                                                                                        | 0b000000 |
| [11:0]         | CCITMIN | Indicates the minimum value that can be programmed in AArch64-TRCCCCTLR.THRESHOLD.<br><br>If AArch64-TRCIDR0.TRCCCI == 1 then the minimum value of this field is 0x001.<br><br>If AArch64-TRCIDR0.TRCCCI == 0 then this field is zero.<br><br><b>0b0000000000100</b> | 0x004    |

## Access

MRS <Xt>, TRCIDR3

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b1011 | 0b111 |

## Accessibility

MRS <Xt>, TRCIDR3

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR3;
    elsif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDR3;
    elsif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR3;

```

### A.12.11 TRCIDR4, ID Register 4

Returns the tracing capabilities of the trace unit.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Trace unit registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0001 | 0001 | 0001 | 0111 | 0000 | xxx0 | 0000 | 0100 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-126: AArch64\_trcidr4 bit assignments

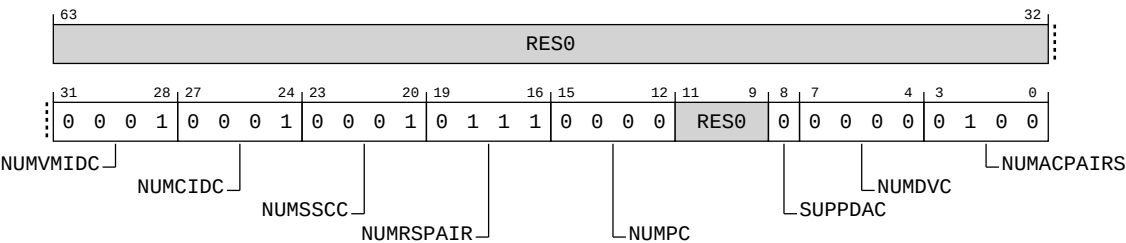


Table A-344: TRCIDR4 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                       | Reset  |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:32] | RES0     | Reserved                                                                                                                                                                          | RES0   |
| [31:28] | NUMVMIDC | Indicates the number of Virtual Context Identifier Comparators that are available for tracing.<br><br>0b0001<br>The implementation has one Virtual Context Identifier Comparator. | 0b0001 |

| Bits    | Name        | Description                                                                                                                                                                                                                                                            | Reset       |
|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [27:24] | NUMCIDC     | Indicates the number of Context Identifier Comparators that are available for tracing.<br><br><b>0b0001</b><br>The implementation has one Context Identifier Comparator.                                                                                               | 0b0001      |
| [23:20] | NUMSSCC     | Indicates the number of Single-shot Comparator Controls that are available for tracing.<br><br><b>0b0001</b><br>The implementation has one Single-shot Comparator Control.                                                                                             | 0b0001      |
| [19:16] | NUMRSPAIR   | Indicates the number of resource selector pairs that are available for tracing.<br><br><b>0b0111</b><br>The implementation has eight resource selector pairs.                                                                                                          | 0b0111      |
| [15:12] | NUMPC       | Indicates the number of PE Comparator Inputs that are available for tracing.<br><br><b>0b0000</b><br>No PE Comparator Inputs are available.                                                                                                                            | 0b0000      |
| [11:9]  | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                               | <b>RES0</b> |
| [8]     | SUPPDAC     | Indicates whether data address comparisons are implemented. Data address comparisons are not implemented in ETE and are reserved for other trace architectures. Allocated in other trace architectures.<br><br><b>0b0</b><br>Data address comparisons not implemented. | 0b0         |
| [7:4]   | NUMDVC      | Indicates the number of data value comparators. Data value comparators are not implemented in ETE and are reserved for other trace architectures. Allocated in other trace architectures.<br><br><b>0b0000</b><br>No data value comparators implemented.               | 0b0000      |
| [3:0]   | NUMACPAIRS  | Indicates the number of Address Comparator pairs that are available for tracing.<br><br><b>0b0100</b><br>The implementation has four Address Comparator pairs.                                                                                                         | 0b0100      |

## Access

MRS <Xt>, TRCIDR4

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b1100 | 0b111 |

## Accessibility

MRS <Xt>, TRCIDR4

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else

```

```
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR4;
elseif PSTATE.EL == EL2 then
    if CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR4;
elseif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCIDR4;
```

A.12.12 TRCIDR5, ID Register 5

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Trace unit registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0010 | 100x | 0100 | 0111 | xxxx | 1001 | 1111 | 1111 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure A-127: AArch64\_trcidr5 bit assignments

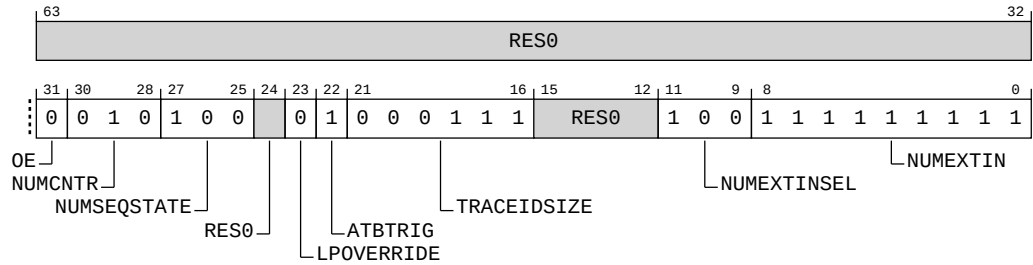


Table A-346: TRCIDR5 bit descriptions

| Bits    | Name        | Description                                                                                                                                                  | Reset      |
|---------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [63:32] | RES0        | Reserved                                                                                                                                                     | RES0       |
| [31]    | OE          | Indicates support for the ETE Trace Output Enable.<br><b>0b0</b><br>ETE Trace Output Enable is not implemented.                                              | 0b0        |
| [30:28] | NUMCNTR     | Indicates the number of Counters that are available for tracing.<br><b>0b010</b><br>Two Counters implemented.                                                | 0b010      |
| [27:25] | NUMSEQSTATE | Indicates if the Sequencer is implemented and the number of Sequencer states that are implemented.<br><b>0b100</b><br>Four Sequencer states are implemented. | 0b100      |
| [24]    | RES0        | Reserved                                                                                                                                                     | RES0       |
| [23]    | LPOVERRIDE  | Indicates support for Low-power Override Mode.<br><b>0b0</b><br>The trace unit does not support Low-power Override Mode.                                     | 0b0        |
| [22]    | ATBTRIG     | Indicates if the implementation can support ATB triggers.<br><b>0b1</b><br>The implementation supports ATB triggers.                                         | 0b1        |
| [21:16] | TRACEIDSIZE | Indicates the trace ID width.<br><b>0b000111</b><br>The implementation supports a 7-bit trace ID.                                                            | 0b000111   |
| [15:12] | RES0        | Reserved                                                                                                                                                     | RES0       |
| [11:9]  | NUMEXTINSEL | Indicates how many External Input Selector resources are implemented.<br><b>0b100</b><br>4 External Input Selector resources are available.                  | 0b100      |
| [8:0]   | NUMEXTIN    | Indicates how many External Inputs are implemented.<br><b>0b11111111</b><br>Unified PMU event selection.                                                     | 0b11111111 |



## Access

MRS &lt;Xt&gt;, TRCIDR5

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b1101 | 0b111 |

## Accessibility

MRS &lt;Xt&gt;, TRCIDR5

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRCID == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR5;
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCIDR5;
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCIDR5;

```

## A.12.13 TRCCIDCVR0, Context Identifier Comparator Value Registers <n>

Contains a Context identifier value.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

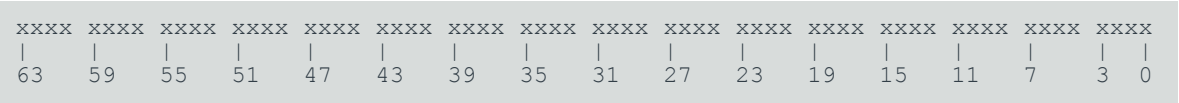
#### Functional group

Trace unit registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-128: AArch64\_trccidcvr0 bit assignments

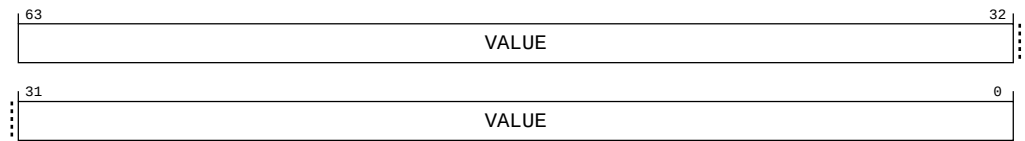


Table A-348: TRCCIDCVR0 bit descriptions

| Bits   | Name  | Description                                                                                                                                                                                                                                               | Reset  |
|--------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:0] | VALUE | Context identifier value. The width of this field is indicated by AArch64-TRCIDR2.CIDSIZE. Unimplemented bits are <b>RES0</b> . After a PE Reset, the trace unit assumes that the Context identifier is zero until the PE updates the Context identifier. | 64 {x} |

Access

Must be programmed if any of the following are true:

- TRCRSCTLR<a>.GROUP == 0b0110 and TRCRSCTLR<a>.CID[n] == 1.
- TRCACATR<a>.CONTEXTTYPE == 0b01 or 0b11 and TRCACATR<a>.CONTEXT == n.

Writes are **CONSTRAINED UNPREDICTABLE** if the trace unit is not in the Idle state.

MRS <Xt>, TRCCIDCVR0

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0011 | 0b0000 | 0b000 |

MSR TRCCIDCVR0, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0011 | 0b0000 | 0b000 |

## Accessibility

Must be programmed if any of the following are true:

- TRCRSCTLR<a>.GROUP == 0b0110 and TRCRSCTLR<a>.CID[n] == 1.
- TRCACATR<a>.CONTEXTTYPE == 0b01 or 0b11 and TRCACATR<a>.CONTEXT == n.

Writes are CONSTRAINED UNPREDICTABLE if the trace unit is not in the Idle state.

MRS <Xt>, TRCCIDCVR0

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.TRC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCCIDCVR[0];
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = TRCCIDCVR[0];
    elseif PSTATE.EL == EL3 then
        if CPTR_EL3.TTA == '1' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = TRCCIDCVR[0];

```

MSR TRCCIDCVR0, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGWTR_EL2.TRC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            TRCCIDCVR[0] = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if CPTR_EL2.TTA == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif CPTR_EL3.TTA == '1' then

```

```

    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCCIDCVR[0] = X[t, 64];
elseif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCCIDCVR[0] = X[t, 64];

```

## A.13 AArch64 Memory Partitioning and Monitoring registers summary

The following summary table provides an overview of all Memory Partitioning and Monitoring registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-351: Memory Partitioning and Monitoring registers summary**

| Name                         | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                   |
|------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------|
| MPAM1_EL1                    | 3   | 0   | C10 | C5  | 0   | See individual bit resets. | 64-bit | MPAM1 Register (EL1)                          |
| MPAM0_EL1                    | 3   | 0   | C10 | C5  | 1   | See individual bit resets. | 64-bit | MPAM0 Register (EL1)                          |
| MPAMHCR_EL2                  | 3   | 4   | C10 | C4  | 0   | See individual bit resets. | 64-bit | MPAM Hypervisor Control Register (EL2)        |
| <a href="#">MPAMVPMV_EL2</a> | 3   | 4   | C10 | C4  | 1   | See individual bit resets. | 64-bit | MPAM Virtual Partition Mapping Valid Register |
| MPAM2_EL2                    | 3   | 4   | C10 | C5  | 0   | See individual bit resets. | 64-bit | MPAM2 Register (EL2)                          |
| <a href="#">MPAMVPM0_EL2</a> | 3   | 4   | C10 | C6  | 0   | See individual bit resets. | 64-bit | MPAM Virtual PARTID Mapping Register 0        |
| <a href="#">MPAMVPM1_EL2</a> | 3   | 4   | C10 | C6  | 1   | See individual bit resets. | 64-bit | MPAM Virtual PARTID Mapping Register 1        |
| MPAM3_EL3                    | 3   | 6   | C10 | C5  | 0   | See individual bit resets. | 64-bit | MPAM3 Register (EL3)                          |

### A.13.1 MPAMVPMV\_EL2, MPAM Virtual Partition Mapping Valid Register

Valid bits for virtual PARTID mapping entries. Each bit *m* corresponds to virtual PARTID mapping entry *m* in the MPAMVPM<*n*>\_EL2 registers where *n* = *m* >> 2.

#### Configurations

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

Functional group

Memory Partitioning and Monitoring registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-129: AArch64\_mpamvpmv\_el2 bit assignments

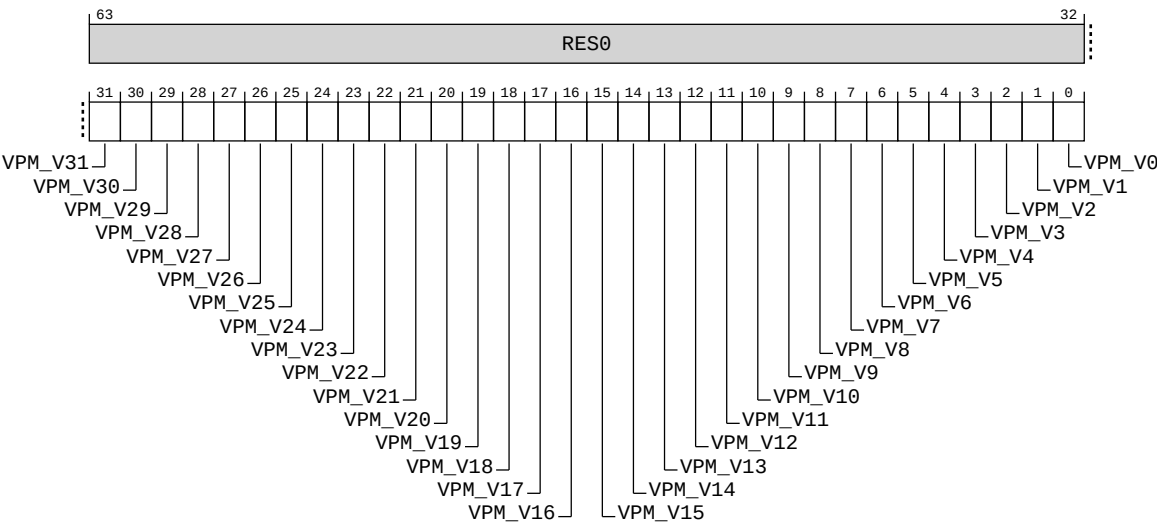


Table A-352: MPAMVPMV\_EL2 bit descriptions

| Bits    | Name                                | Description                                                                             | Reset  |
|---------|-------------------------------------|-----------------------------------------------------------------------------------------|--------|
| [63:32] | RES0                                | Reserved                                                                                | RES0   |
| [31:0]  | VPM_V<m>, bit[m], where m = 31 to 0 | Contains valid bit for virtual PARTID mapping entry corresponding to virtual PARTID<m>. | 32 {x} |

Access

MRS <Xt>, MPAMVPMV\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0100 | 0b001 |

MSR MPAMVPMV\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0100 | 0b001 |

Accessibility

MRS <Xt>, MPAMVPMV\_EL2

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = MPAMVPMV_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = MPAMVPMV_EL2;
```

MSR MPAMVPMV\_EL2, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        MPAMVPMV_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    MPAMVPMV_EL2 = X[t, 64];
```

A.13.2 MPAMVPMO\_EL2, MPAM Virtual PARTID Mapping Register 0

MPAMVPMO\_EL2 provides mappings from virtual PARTIDs 0 - 3 to physical PARTIDs.

AArch64-MPAMIDR\_EL1.VPMR\_MAX field gives the index of the highest implemented MPAMVPM<n>\_EL2 register. VPMR\_MAX can be as large as 7 (8 registers) or 32 virtual PARTIDs. If AArch64-MPAMIDR\_EL1.VPMR\_MAX == 0, there is only a single MPAMVPM<n>\_EL2 register, AArch64-MPAMVPMO\_EL2.

Virtual PARTID mapping is enabled by AArch64-MPAMHCR\_EL2.EL1\_VPMEN for PARTIDs in AArch64-MPAM1\_EL1 and by AArch64-MPAMHCR\_EL2.ELO\_VPMEN for PARTIDs in AArch64-MPAMO\_EL1.

A virtual-to-physical PARTID mapping entry, PhyPARTID<n>, is valid only when the AArch64-MPAMVPMV\_EL2.VPM\_V bit in bit position n is set to 1.

Configurations

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

Width

64

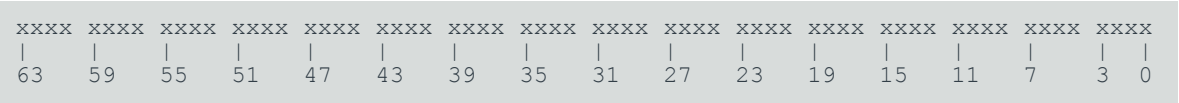
Functional group

Memory Partitioning and Monitoring registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-130: AArch64\_mpamvpm0\_el2 bit assignments

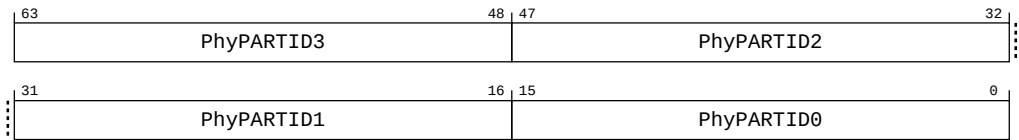


Table A-355: MPAMVPM0\_EL2 bit descriptions

| Bits    | Name       | Description                                                                                                               | Reset  |
|---------|------------|---------------------------------------------------------------------------------------------------------------------------|--------|
| [63:48] | PhyPARTID3 | Virtual PARTID Mapping Entry for virtual PARTID 3. PhyPARTID3 gives the mapping of virtual PARTID 3 to a physical PARTID. | 16 {x} |
| [47:32] | PhyPARTID2 | Virtual PARTID Mapping Entry for virtual PARTID 2. PhyPARTID2 gives the mapping of virtual PARTID 2 to a physical PARTID. | 16 {x} |
| [31:16] | PhyPARTID1 | Virtual PARTID Mapping Entry for virtual PARTID 1. PhyPARTID1 gives the mapping of virtual PARTID 1 to a physical PARTID. | 16 {x} |

| Bits   | Name       | Description                                                                                                               | Reset |
|--------|------------|---------------------------------------------------------------------------------------------------------------------------|-------|
| [15:0] | PhyPARTID0 | Virtual PARTID Mapping Entry for virtual PARTID 0. PhyPARTID0 gives the mapping of virtual PARTID 0 to a physical PARTID. | 16{x} |

## Access

MRS <Xt>, MPAMVPMO\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0110 | 0b000 |

MSR MPAMVPMO\_EL2, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0110 | 0b000 |

## Accessibility

MRS <Xt>, MPAMVPMO\_EL2

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = MPAMVPMO_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = MPAMVPMO_EL2;

```

MSR MPAMVPMO\_EL2, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        MPAMVPMO_EL2 = X[t, 64];
elseif PSTATE.EL == EL3 then
    MPAMVPMO_EL2 = X[t, 64];

```



### A.13.3 MPAMVPM1\_EL2, MPAM Virtual PARTID Mapping Register 1

MPAMVPM1\_EL2 provides mappings from virtual PARTIDs 4 - 7 to physical PARTIDs.

AArch64-MPAMIDR\_EL1.VPMR\_MAX field gives the index of the highest implemented AArch64-MPAMVPM0\_EL2 to AArch64-MPAMVPM7\_EL2 registers. VPMR\_MAX can be as large as 7 (8 registers) or 32 virtual PARTIDs. If AArch64-MPAMIDR\_EL1.VPMR\_MAX == 0, there is only a single MPAMVPM<n>\_EL2 register, AArch64-MPAMVPM0\_EL2.

Virtual PARTID mapping is enabled by AArch64-MPAMHCR\_EL2.EL1\_VPMEN for PARTIDs in AArch64-MPAM1\_EL1 and by MPAMHCR\_EL2.ELO\_VPMEN for PARTIDs in AArch64-MPAMO\_EL1.

A virtual-to-physical PARTID mapping entry, PhyPARTID<n>, is valid only when the AArch64-MPAMVPMV\_EL2.VPM\_V bit in bit position n is set to 1.

#### Configurations

This register has no effect if EL2 is not enabled in the current Security state.

#### Attributes

##### Width

64

##### Functional group

Memory Partitioning and Monitoring registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 0    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-131: AArch64\_mpamvpm1\_el2 bit assignments

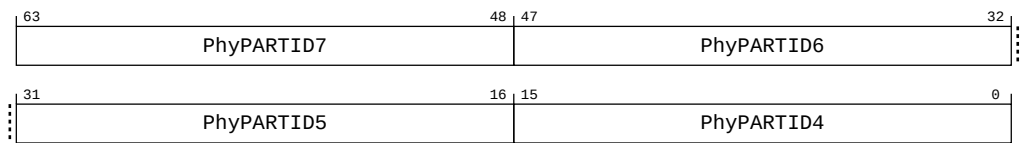


**Table A-358: MPAMVPM1\_EL2 bit descriptions**

| Bits    | Name       | Description                                                                                                               | Reset |
|---------|------------|---------------------------------------------------------------------------------------------------------------------------|-------|
| [63:48] | PhyPARTID7 | Virtual PARTID Mapping Entry for virtual PARTID 7. PhyPARTID7 gives the mapping of virtual PARTID 7 to a physical PARTID. | 16{x} |
| [47:32] | PhyPARTID6 | Virtual PARTID Mapping Entry for virtual PARTID 6. PhyPARTID6 gives the mapping of virtual PARTID 6 to a physical PARTID. | 16{x} |
| [31:16] | PhyPARTID5 | Virtual PARTID Mapping Entry for virtual PARTID 5. PhyPARTID5 gives the mapping of virtual PARTID 5 to a physical PARTID. | 16{x} |
| [15:0]  | PhyPARTID4 | Virtual PARTID Mapping Entry for virtual PARTID 4. PhyPARTID4 gives the mapping of virtual PARTID 4 to a physical PARTID. | 16{x} |

### Access

MRS &lt;Xt&gt;, MPAMVPM1\_EL2

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0110 | 0b001 |

MSR MPAMVPM1\_EL2, &lt;Xt&gt;

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0110 | 0b001 |

### Accessibility

MRS &lt;Xt&gt;, MPAMVPM1\_EL2

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = MPAMVPM1_EL2;
elseif PSTATE.EL == EL3 then
    X[t, 64] = MPAMVPM1_EL2;

```

MSR MPAMVPM1\_EL2, &lt;Xt&gt;

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    UNDEFINED;
elseif PSTATE.EL == EL2 then
    if MPAM3_EL3.TRAPLOWER == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        MPAMVPM1_EL2 = X[t, 64];

```

```
elseif PSTATE.EL == EL3 then
    MPAMVPM1_EL2 = X[t, 64];
```

## A.14 AArch64 RAS registers summary

The following summary table provides an overview of all RAS registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-361: RAS registers summary**

| Name                          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                         |
|-------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|-----------------------------------------------------|
| <a href="#">ERRIDR_EL1</a>    | 3   | 0   | C5  | C3  | 0   | See individual bit resets. | 64-bit | Error Record ID Register                            |
| <a href="#">ERRSELR_EL1</a>   | 3   | 0   | C5  | C3  | 1   | See individual bit resets. | 64-bit | Error Record Select Register                        |
| <a href="#">ERXFR_EL1</a>     | 3   | 0   | C5  | C4  | 0   | See individual bit resets. | 64-bit | Selected Error Record Feature Register              |
| <a href="#">ERXCTLR_EL1</a>   | 3   | 0   | C5  | C4  | 1   | See individual bit resets. | 64-bit | Selected Error Record Control Register              |
| <a href="#">ERXSTATUS_EL1</a> | 3   | 0   | C5  | C4  | 2   | See individual bit resets. | 64-bit | Selected Error Record Primary Status Register       |
| <a href="#">ERXADDR_EL1</a>   | 3   | 0   | C5  | C4  | 3   | See individual bit resets. | 64-bit | Selected Error Record Address Register              |
| <a href="#">ERXPFGF_EL1</a>   | 3   | 0   | C5  | C4  | 4   | See individual bit resets. | 64-bit | Selected Pseudo-fault Generation Feature Register   |
| <a href="#">ERXPFGCTL_EL1</a> | 3   | 0   | C5  | C4  | 5   | See individual bit resets. | 64-bit | Selected Pseudo-fault Generation Control Register   |
| <a href="#">ERXPFGCDN_EL1</a> | 3   | 0   | C5  | C4  | 6   | See individual bit resets. | 64-bit | Selected Pseudo-fault Generation Countdown Register |
| <a href="#">ERXMISCO_EL1</a>  | 3   | 0   | C5  | C5  | 0   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 0      |
| <a href="#">ERXMISC1_EL1</a>  | 3   | 0   | C5  | C5  | 1   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 1      |
| <a href="#">ERXMISC2_EL1</a>  | 3   | 0   | C5  | C5  | 2   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 2      |
| <a href="#">ERXMISC3_EL1</a>  | 3   | 0   | C5  | C5  | 3   | See individual bit resets. | 64-bit | Selected Error Record Miscellaneous Register 3      |
| <a href="#">DISR_EL1</a>      | 3   | 0   | C12 | C1  | 1   | See individual bit resets. | 64-bit | Deferred Interrupt Status Register                  |

| Name      | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                |
|-----------|-----|-----|-----|-----|-----|----------------------------|--------|--------------------------------------------|
| VSESR_EL2 | 3   | 4   | C5  | C2  | 3   | See individual bit resets. | 64-bit | Virtual SError Exception Syndrome Register |
| VDISR_EL2 | 3   | 4   | C12 | C1  | 1   | See individual bit resets. | 64-bit | Virtual Deferred Interrupt Status Register |

A.14.1 ERRIDR\_EL1, Error Record ID Register

Defines the highest numbered index of the error records that can be accessed through the Error Record System registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

RAS registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0010 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-132: AArch64\_erridr\_el1 bit assignments

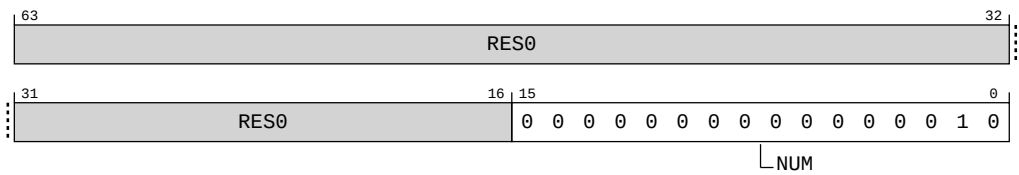


**Table A-362: ERRIDR\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                   | Reset  |
|---------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:16] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                      | RES0   |
| [15:0]  | NUM  | <p>Highest numbered index of the records that can be accessed through the Error Record System registers plus one. Zero indicates no records can be accessed through the Error Record System registers.</p> <p>Each implemented record is owned by a node. A node might own multiple records.</p> <p><b>0b0000000000000010</b></p> <p>Two Records Present.</p> | 0x0002 |

### Access

MRS &lt;Xt&gt;, ERRIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0011 | 0b000 |

### Accessibility

MRS &lt;Xt&gt;, ERRIDR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERRIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERRIDR_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERRIDR_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERRIDR_EL1;

```

## A.14.2 ERRSELR\_EL1, Error Record Select Register

Selects an error record to be accessed through the Error Record System registers.

### Configurations

If AArch64-ERRIDR\_EL1 indicates that zero error records are implemented, then it is IMPLEMENTATION DEFINED whether ERRSELR\_EL1 is UNDEFINED or RES0.

Attributes

Width

64

Functional group

RAS registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-133: AArch64\_errselr\_el1 bit assignments

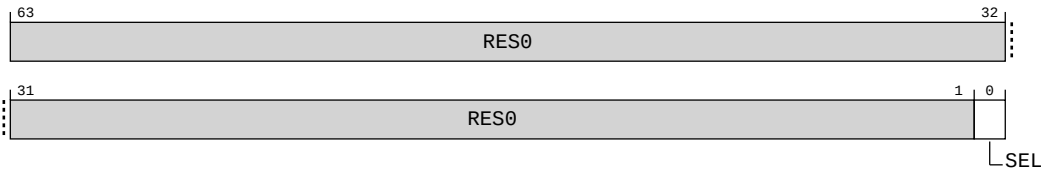


Table A-364: ERRSELR\_EL1 bit descriptions

| Bits   | Name | Description                                                                                                                             | Reset |
|--------|------|-----------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:1] | RES0 | Reserved                                                                                                                                | RES0  |
| [0]    | SEL  | <b>0b0</b><br>Selects record 0, containing errors from DSU RAMs<br><br><b>0b1</b><br>Selects record 1, containing errors from Core RAMs | 0b0   |

Access

MRS <Xt>, ERRSELR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0011 | 0b001 |

MSR ERRSELR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0011 | 0b001 |

## Accessibility

MRS <Xt>, ERRSELR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERRSELR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERRSELR_EL1;
    elsif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERRSELR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = ERRSELR_EL1;

```

MSR ERRSELR\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERRSELR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERRSELR_EL1 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ERRSELR_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        ERRSELR_EL1 = X[t, 64];

```

A.14.3 ERXFR\_EL1, Selected Error Record Feature Register

Accesses ext-ERR<n>FR for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

RAS registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | x101 | 0001 | xxxx | xxxx | xxxx | xxxx | 1xxx | xx00 | 0001 | 0000 | 1010 | 1001 | 1010 | xx10 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-134: AArch64\_erxfr\_el1 bit assignments

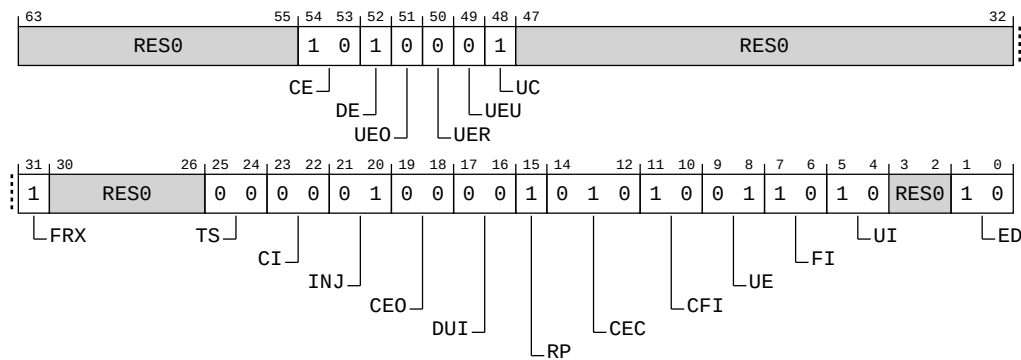


Table A-367: ERXFR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:55] | RES0 | Reserved    | RES0  |



| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [54:53] | CE   | Corrected Error recording. Describes the types of Corrected errors the node can record, if any.<br><b>0b10</b><br>Records only non-specific Corrected errors. That is, Corrected errors recorded by setting ERXSTATUS_EL1.CE to 0b10.                                                                                                                                                        | 0b10  |
| [52]    | DE   | Deferred Error recording. Describes whether the node supports recording Deferred errors.<br><b>0b1</b><br>Records Deferred errors.                                                                                                                                                                                                                                                           | 0b1   |
| [51]    | UEO  | Latent or Restartable Error recording. Describes whether the node supports recording Latent or Restartable errors.<br><b>0b0</b><br>Does not record Latent or Restartable errors.                                                                                                                                                                                                            | 0b0   |
| [50]    | UER  | Signaled or Recoverable Error recording. Describes whether the node supports recording Signaled or Recoverable errors.<br><b>0b0</b><br>Does not record Signaled or Recoverable errors.                                                                                                                                                                                                      | 0b0   |
| [49]    | UEU  | Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.<br><b>0b0</b><br>Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.                                                                                                                                                                       | 0b0   |
| [48]    | UC   | Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.<br><b>0b1</b><br>Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.                                                                                                                                                                       | 0b1   |
| [47:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                     | RES0  |
| [31]    | FRX  | Feature Register extension. Defines whether ERXFR_EL1[63:48] are architecturally defined.<br><b>0b1</b><br>ERXFR_EL1[63:48] are defined by the architecture.                                                                                                                                                                                                                                 | 0b1   |
| [30:26] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                     | RES0  |
| [25:24] | TS   | Timestamp Extension. Indicates whether, for each error record <m> owned by this node, ERXMISC3_EL1 is used as the timestamp register, and, if it is, the timebase used by the timestamp.<br><b>0b00</b><br>The node does not support a timestamp register.                                                                                                                                   | 0b00  |
| [23:22] | CI   | Critical error interrupt. Indicates whether the critical error interrupt and associated controls are implemented.<br><b>0b00</b><br>Does not support the critical error interrupt. ERXCTLR_EL1.CI is <b>RES0</b> .                                                                                                                                                                           | 0b00  |
| [21:20] | INJ  | Fault Injection Extension. Indicates whether the RAS Common Fault Injection Model Extension is implemented.<br><b>0b01</b><br>The node implements the RAS Common Fault Injection Model Extension. See ERXPFGF_EL1 for more information.                                                                                                                                                      | 0b01  |
| [19:18] | CEO  | Corrected Error overwrite. Indicates the behavior when a second Corrected error is detected after a first Corrected error has been recorded by an error record <m> owned by the node.<br><b>0b00</b><br>Counts Corrected errors if a counter is implemented. Keeps the previous error syndrome. If the counter overflows, or no counter is implemented, then ERXSTATUS_EL1.OF is set to 0b1. | 0b00  |

| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                              | Reset       |
|---------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [17:16] | DUI         | Error recovery interrupt for deferred errors control. Indicates whether the control for enabling error recovery interrupts on deferred errors are implemented.<br><br><b>0b00</b><br>Does not support the control for enabling error recovery interrupts on deferred errors. ERXCTLR_EL1.DUI is <b>RES0</b> .                                                            | 0b00        |
| [15]    | RP          | Repeat counter. Indicates whether the node implements the repeat Corrected error counter in ERXMISCO_EL1 for each error record <m> owned by the node that implements the standard Corrected error counter.<br><br><b>0b1</b><br>A first (repeat) counter and a second (other) counter are implemented. The repeat counter is the same size as the primary error counter. | 0b1         |
| [14:12] | CEC         | Corrected Error Counter. Indicates whether the node implements the standard Corrected error counter (CE counter) mechanisms in ERXMISCO_EL1 for each error record <m> owned by the node that can record countable errors.<br><br><b>0b010</b><br>Implements an 8-bit Corrected error counter in ERXMISCO_EL1[39:32].                                                     | 0b010       |
| [11:10] | CFI         | Fault handling interrupt for corrected errors. Indicates whether the control for enabling fault handling interrupts on corrected errors are implemented.<br><br><b>0b10</b><br>Control for enabling fault handling interrupts on corrected errors is supported and controllable using ERXCTLR_EL1.CFI.                                                                   | 0b10        |
| [9:8]   | UE          | In-band uncorrected error reporting. Indicates whether the in-band uncorrected error reporting (External Aborts) and associated controls are implemented.<br><br><b>0b01</b><br>In-band uncorrected error reporting (External Aborts) is supported and always enabled. ERXCTLR_EL1.UE is <b>RES0</b> .                                                                   | 0b01        |
| [7:6]   | FI          | Fault handling interrupt. Indicates whether the fault handling interrupt and associated controls are implemented.<br><br><b>0b10</b><br>Fault handling interrupt is supported and controllable using ERXCTLR_EL1.FI.                                                                                                                                                     | 0b10        |
| [5:4]   | UI          | Error recovery interrupt for uncorrected errors. Indicates whether the error handling interrupt and associated controls are implemented.<br><br><b>0b10</b><br>Error handling interrupt is supported and controllable using ERXCTLR_EL1.UI.                                                                                                                              | 0b10        |
| [3:2]   | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                 | <b>RES0</b> |
| [1:0]   | ED          | Error reporting and logging. Indicates whether error record <n> is the first record owned the node, and, if so, whether it implements the controls for enabling and disabling error reporting and logging.<br><br><b>0b10</b><br>Error reporting and logging is controllable using ERXCTLR_EL1.ED.                                                                       | 0b10        |

## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXFR\_EL1 is RAZ.
- Direct reads of ERXFR\_EL1 are NOPs.

- Direct reads of ERXFR\_EL1 are UNDEFINED.

MRS <Xt>, ERXFR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b000 |

### Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXFR\_EL1 is RAZ.
- Direct reads of ERXFR\_EL1 are NOPs.
- Direct reads of ERXFR\_EL1 are UNDEFINED.

MRS <Xt>, ERXFR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXFR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXFR_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXFR_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERXFR_EL1;

```

## A.14.4 ERXCTLR\_EL1, Selected Error Record Control Register

Accesses ext-ERR<n>CTLR for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

Functional group

RAS registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | xxxx | 00x0 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-135: AArch64\_erxctlr\_el1 bit assignments

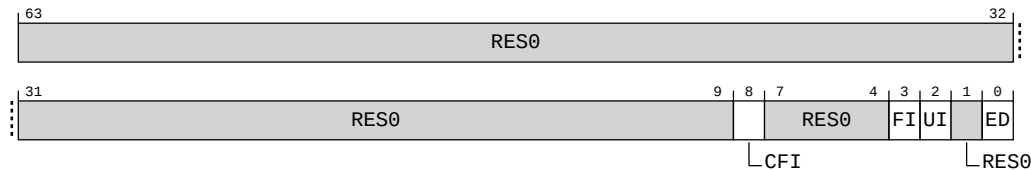


Table A-369: ERXCTLR\_EL1 bit descriptions

| Bits   | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|--------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:9] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RES0  |
| [8]    | CFI  | <p>Fault handling interrupt for Corrected errors enable.</p> <p>This control applies to errors arising from both reads and writes.</p> <p>The fault handling interrupt is generated when one of the standard CE counters on ERXMISCO_EL1 overflows and the overflow bit is set. The possible values are:</p> <p><b>0b0</b></p> <p>Fault handling interrupt not generated for Corrected errors.</p> <p><b>0b1</b></p> <p>Fault handling interrupt generated for Corrected errors.</p> <p>The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.</p> <p>Cold reset only. Unaffected by Warm reset</p> | 0b0   |
| [7:4]  | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RES0  |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [3]  | FI   | <p>Fault handling interrupt enable.</p> <p>This control applies to errors arising from both reads and writes.</p> <p>The fault handling interrupt is generated for all detected Deferred errors and Uncorrected errors. The possible values are:</p> <p><b>0b0</b></p> <p>Fault handling interrupt disabled.</p> <p><b>0b1</b></p> <p>Fault handling interrupt enabled.</p> <p>The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.</p> <p>Cold reset only. Unaffected by Warm reset</p>    | 0b0   |
| [2]  | UI   | <p>Uncorrected error recovery interrupt enable.</p> <p>This control applies to errors arising from both reads and writes.</p> <p>When enabled, the error recovery interrupt is generated for all detected Uncorrected errors that are not deferred.</p> <p><b>0b0</b></p> <p>Error recovery interrupt disabled.</p> <p><b>0b1</b></p> <p>Error recovery interrupt enabled.</p> <p>The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.</p> <p>Cold reset only. Unaffected by Warm reset</p> | 0b0   |
| [1]  | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [0]  | ED   | <p>Error Detection and correction enable. The possible values are:</p> <p><b>0b0</b></p> <p>Error detection and correction disabled.</p> <p><b>0b1</b></p> <p>Error detection and correction enabled.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                                                                                                                                                                                  | 0b0   |

## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXCTLR\_EL1 is RAZ/WI.
- Direct reads and writes of ERXCTLR\_EL1 are NOPs.
- Direct reads and writes of ERXCTLR\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>CTLR is not present, meaning reads and writes of ERXCTLR\_EL1 are **RES0**.

MRS <Xt>, ERXCTLR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b001 |

MSR ERXCTLR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b001 |

### Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXCTLR\_EL1 is RAZ/WI.
- Direct reads and writes of ERXCTLR\_EL1 are NOPs.
- Direct reads and writes of ERXCTLR\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>CTLR is not present, meaning reads and writes of ERXCTLR\_EL1 are RES0.

MRS <Xt>, ERXCTLR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXCTLR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXCTLR_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXCTLR_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERXCTLR_EL1;

```

MSR ERXCTLR\_EL1, <Xt>

```

if PSTATE.EL == EL0 then

```

```

    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXCTLR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXCTLR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ERXCTLR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL3 then
        ERXCTLR_EL1 = X[t, 64];

```

### A.14.5 ERXSTATUS\_EL1, Selected Error Record Primary Status Register

Accesses ext-ERR<n>STATUS for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

RAS registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | xxxx | xxxx | xxxx | xxx0 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-136: AArch64\_erxstatus\_el1 bit assignments**

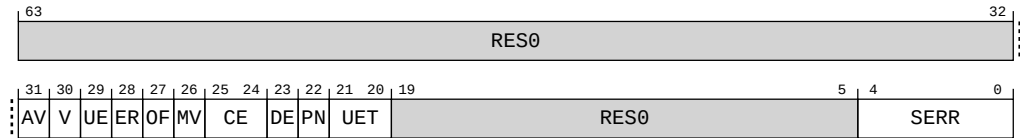


**Table A-372: ERXSTATUS\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0  |
| [31]    | AV   | Address Valid. The possible values are:<br><br><b>0b0</b><br>ERXADDR_EL1 not valid.<br><br><b>0b1</b><br>ERXADDR_EL1 contains an address associated with the highest priority error recorded by this record.<br><br>This bit is read/write-one-to-clear.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                                                                                                                                                | 0b0   |
| [30]    | V    | Status Register Valid. The possible values are:<br><br><b>0b0</b><br>ERXSTATUS_EL1 not valid.<br><br><b>0b1</b><br>ERXSTATUS_EL1 valid. At least one error has been recorded.<br><br>This bit is read/write-one-to-clear.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                                                                                                                                                                               | 0b0   |
| [29]    | UE   | Uncorrected Error. The possible values are:<br><br><b>0b0</b><br>No errors have been detected, or all detected errors have been either corrected or deferred.<br><br><b>0b1</b><br>At least one detected error was not corrected and not deferred.<br><br>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.<br><br>This bit is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.<br><br>This bit is read/write-one-to-clear.<br><br>Cold reset only. Unaffected by Warm reset | 0b0   |



| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [28] | ER   | <p>Error Reported. The possible values are:</p> <p><b>0b0</b></p> <p>No in-band error (External Abort) reported.</p> <p><b>0b1</b></p> <p>An External Abort was signaled by the node to the master making the access or other transaction.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p> <p><b>Note:</b><br/>An External Abort signaled by the node might be masked and not generate any exception.</p>                                                                                                                                                                                                                                                                                                                                            | 0b0   |
| [27] | OF   | <p>Overflow. The possible values are:</p> <p><b>0b0</b></p> <p>If UE == 1, then no error status for an Uncorrected error has been discarded.</p> <p>If UE == 0 and DE == 1, then no error status for a Deferred error has been discarded.</p> <p>If UE == 0, DE == 0, and CE != 0b00, then the corrected error counter has not overflowed.</p> <p><b>0b1</b></p> <p>More than one error has occurred and so details of the other error have been discarded.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>Cold reset only. Unaffected by Warm reset</p> <p>This bit is read/write-one-to-clear.</p> | 0b0   |
| [26] | MV   | <p>Miscellaneous Registers Valid. The possible values are:</p> <p><b>0b0</b></p> <p>ERXMISC&lt;m&gt;_EL1 not valid.</p> <p><b>0b1</b></p> <p>This bit indicates that the ERXMISC&lt;m&gt;_EL1 registers contain additional information for an error recorded by this record.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p> <p><b>Note:</b><br/>If the ERXMISC&lt;m&gt;_EL1 registers can contain additional information for a previously recorded error, then the contents must be self-describing to software or a user. For example, certain fields might relate only to Corrected errors, and other fields only to the most recent error that was not discarded.</p>                                                                            | 0b0   |

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [25:24] | CE   | <p>Corrected Error. The possible values are:</p> <p><b>0b00</b><br/>No errors were corrected.</p> <p><b>0b01</b><br/>At least one transient error was corrected.</p> <p><b>0b10</b><br/>At least one error was corrected.</p> <p><b>0b11</b><br/>At least one persistent error was corrected.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this field is nonzero, then Arm recommends that software write ones to this field to clear this field to zero.</p> <p>This field is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>This field is read/write-ones-to-clear. Writing a value other than all-zeros or all-ones sets this field to an <b>UNKNOWN</b> value.</p> <p>Cold reset only. Unaffected by Warm reset</p> | 0b00  |
| [23]    | DE   | <p>Deferred Error. The possible values are:</p> <p><b>0b0</b><br/>No errors were deferred.</p> <p><b>0b1</b><br/>At least one error was not corrected and deferred.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                                                                                                    | 0b0   |
| [22]    | PN   | <p>Poison. The value is:</p> <p><b>0b0</b><br/>This core cannot distinguish a poisoned value from a corrupted value.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if any of the following are true:</p> <ul style="list-style-type: none"> <li>ERXSTATUS_EL1.V == 0b0.</li> <li>ERXSTATUS_EL1.{DE,UE} == {0,0}.</li> </ul> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                         | 0b0   |

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset    |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [21:20] | UET  | Uncorrected Error Type. The value is:<br><br><b>0b00</b><br>Uncorrected error, Uncontainable error (UC).<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0b00     |
| [19:5]  | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RES0     |
| [4:0]   | SERR | Primary error code.<br><br>The primary error code might be used by a fault handling agent to triage an error without requiring device-specific code. For example, to count and threshold corrected errors in software, or generate a short log entry.<br><br>The possible values are:<br><b>0b000000</b><br>No error<br><b>0b000010</b><br>ECC error from internal data buffer.<br><b>0b000110</b><br>ECC error on cache data RAM.<br><b>0b000111</b><br>ECC error on cache tag or dirty RAM.<br><b>0b010000</b><br>Parity error on TLB data RAM.<br><b>0b100010</b><br>Error response for a cache copyback.<br><b>0b101011</b><br>Deferred error from slave not supported at the consumer. For example, poisoned data received from a slave by a master that cannot defer the error further.<br><br>Cold reset only. Unaffected by Warm reset | 0b000000 |

## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXSTATUS\_EL1 is RAZ/WI.
- Direct reads and writes of ERXSTATUS\_EL1 are NOPs.
- Direct reads and writes of ERXSTATUS\_EL1 are UNDEFINED.

ext-ERR<n>STATUS describes additional constraints that also apply when ext-ERR<n>STATUS is accessed through ERXSTATUS\_EL1.

MRS <Xt>, ERXSTATUS\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b010 |

MSR ERXSTATUS\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b010 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXSTATUS\_EL1 is RAZ/WI.
- Direct reads and writes of ERXSTATUS\_EL1 are NOPs.
- Direct reads and writes of ERXSTATUS\_EL1 are UNDEFINED.

ext-ERR<n>STATUS describes additional constraints that also apply when ext-ERR<n>STATUS is accessed through ERXSTATUS\_EL1.

MRS <Xt>, ERXSTATUS\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXSTATUS_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXSTATUS_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXSTATUS_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERXSTATUS_EL1;

```

MSR ERXSTATUS\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXSTATUS_EL1 == '1'
    then

```

```
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXSTATUS_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ERXSTATUS_EL1 = X[t, 64];
    elseif PSTATE.EL == EL3 then
        ERXSTATUS_EL1 = X[t, 64];
```

A.14.6 ERXADDR\_EL1, Selected Error Record Address Register

Accesses ext-ERR<n>ADDR for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

RAS registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-137: AArch64\_erxaddr\_el1 bit assignments

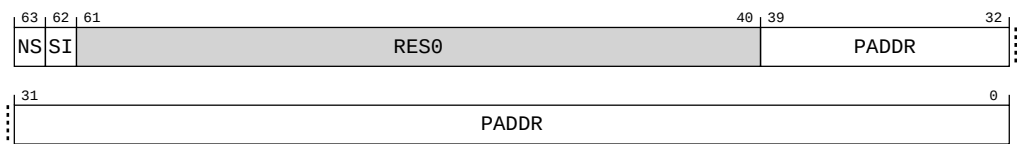


Table A-375: ERXADDR\_EL1 bit descriptions

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                           | Reset  |
|---------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63]    | NS    | Non-secure attribute.<br><br><b>0b0</b><br>ERR<n>ADDR.PADDR is a Secure address.<br><br><b>0b1</b><br>ERR<n>ADDR.PADDR is a Non-secure address.                                                                                                                                                                                                                       | x      |
| [62]    | SI    | Secure Incorrect. Indicates whether ERR<n>ADDR.NS is valid.<br><br><b>0b0</b><br>ERR<n>ADDR.NS is correct. That is, it matches the programmers' view of the Non-secure attribute for the recorded location.<br><br><b>0b1</b><br>ERR<n>ADDR.NS might not be correct, and might not match the programmers' view of the Non-secure attribute for the recorded location. | x      |
| [61:40] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                              | RES0   |
| [39:0]  | PADDR | Physical Address [39:0]. Address of the recorded location                                                                                                                                                                                                                                                                                                             | 40 {x} |

Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXADDR\_EL1 is RAZ/WI.
- Direct reads and writes of ERXADDR\_EL1 are NOPs.
- Direct reads and writes of ERXADDR\_EL1 are UNDEFINED.

ext-ERR<n>ADDR describes additional constraints that also apply when ext-ERR<n>ADDR is accessed through ERXADDR\_EL1.

MRS <Xt>, ERXADDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b011 |

MSR ERXADDR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b011 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXADDR\_EL1 is RAZ/WI.
- Direct reads and writes of ERXADDR\_EL1 are NOPs.
- Direct reads and writes of ERXADDR\_EL1 are UNDEFINED.

ext-ERR<n>ADDR describes additional constraints that also apply when ext-ERR<n>ADDR is accessed through ERXADDR\_EL1.

MRS <Xt>, ERXADDR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXADDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXADDR_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXADDR_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERXADDR_EL1;

```

MSR ERXADDR\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXADDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXADDR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then

```

```
if SCR_EL3.TERR == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ERXADDR_EL1 = X[t, 64];
elseif PSTATE.EL == EL3 then
    ERXADDR_EL1 = X[t, 64];
```

A.14.7 ERXPFGF\_EL1, Selected Pseudo-fault Generation Feature Register

Accesses ext-ERR<n>PFGF for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

RAS registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | x100 | xxxx | xxxx | xxxx | xxx0 | 0000 | 0110 | 0010 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-138: AArch64\_erxpfgf\_el1 bit assignments

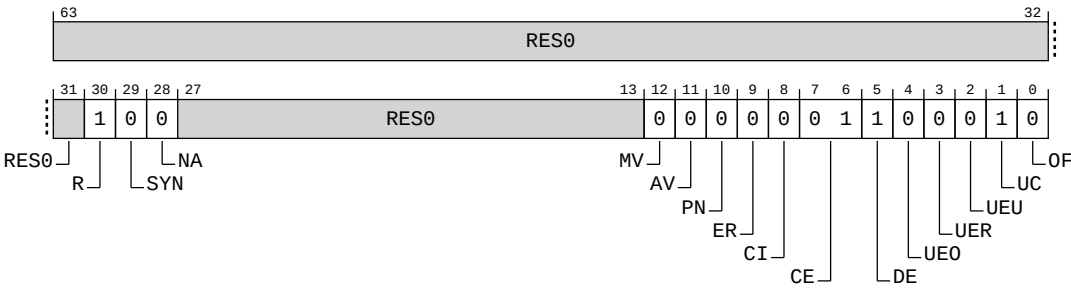




**Table A-378: ERXPFGF\_EL1 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:31] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES0  |
| [30]    | R    | Restartable bit. When it reaches zero, the Error Generation Counter restarts from the ERXPFGCDN_EL1 value or stops. The value is:<br><br><b>0b1</b><br><br>Error Generation Counter restart mode is implemented and is controlled by ERXPFGCTL_EL1.R. ERXPFGCTL_EL1.R is a read/write field                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0b1   |
| [29]    | SYN  | Syndrome. Fault syndrome injection. The value is:<br><br><b>0b0</b><br><br>When an injected error is recorded, the node sets ext-ERR<n>STATUS.{IERR, SERR} to IMPLEMENTATION DEFINED values. ext-ERR<n>STATUS.{IERR, SERR} are <b>UNKNOWN</b> when ext-ERR<n>STATUS.V is 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0b0   |
| [28]    | NA   | No access required. Defines whether this component fakes detection of the error on an access to the component or spontaneously in the fault injection state.<br><br><b>0b0</b><br><br>The component fakes detection of the error on an access to the component.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0b0   |
| [27:13] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES0  |
| [12]    | MV   | Miscellaneous syndrome.<br><br>Defines whether software can control all or part of the syndrome recorded in the ERR<n>MISC<m> registers when an injected error is recorded.<br><br><b>0b0</b><br><br>When an injected error is recorded, the node might update the ERR<n>MISC<m> registers:<br><ul style="list-style-type: none"> <li>If any syndrome is recorded by the node in the ERR&lt;n&gt;MISC&lt;m&gt; registers, then ext-ERR&lt;n&gt;STATUS.MV is set to 1.</li> <li>Otherwise, ext-ERR&lt;n&gt;STATUS.MV is unchanged.</li> </ul> <b>Note:</b><br>If ERXPFGF_EL1.MV == 0b1, software can write specific values into the ERXMISC<m>_EL1 registers when setting up a fault injection event. The values that can be written to these registers are <b>IMPLEMENTATION DEFINED</b> . | 0b0   |
| [11]    | AV   | Address syndrome. Address syndrome injection. The value is:<br><br><b>0b0</b><br><br>When an injected error is recorded, the node might record an address in ext-ERR<n>ADDR. If an address is recorded in ext-ERR<n>ADDR, then ext-ERR<n>STATUS.AV is set to 1. Otherwise, ext-ERR<n>ADDR and ext-ERR<n>STATUS.AV are unchanged.                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0b0   |
| [10]    | PN   | Poison flag. Describes how the fault generation feature of the node sets the ERXSTATUS_EL1.PN status flag. The value is:<br><br><b>0b0</b><br><br>When an injected error is recorded, the node sets ERXSTATUS_EL1.PN to 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0b0   |
| [9]     | ER   | Error Reported flag. Describes how the fault generation feature of the node sets the ERXSTATUS_EL1.ER status flag. The value is:<br><br><b>0b0</b><br><br>When an injected error is recorded, the node sets ext-ERR<n>STATUS.ER according to the architecture-defined rules for setting the ER field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0b0   |

| Bits  | Name | Description                                                                                                                                                                                                                                                                                                                          | Reset |
|-------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [8]   | CI   | Critical Error flag. Describes how the fault generation feature of the node sets the ERXSTATUS_EL1.CI status flag. The value is:<br><br><b>0b0</b><br><br>The node does not support this type of flag<br><br>This behavior replaces the architecture-defined rules for setting the CI bit.                                           | 0b0   |
| [7:6] | CE   | Corrected Error generation. The value is:<br><br><b>0b01</b><br><br>The fault generation feature of the node allows generation of a non-specific Corrected Error, that is, a Corrected Error that is recorded as ERXSTATUS_EL1.CE == 0b10.<br><br>All other values are reserved.                                                     | 0b01  |
| [5]   | DE   | Deferred Error generation. The value is:<br><br><b>0b1</b><br><br>The fault generation feature of the node allows generation of this type of error.                                                                                                                                                                                  | 0b1   |
| [4]   | UEO  | Latent or Restartable Error generation. The value is:<br><br><b>0b0</b><br><br>The fault generation feature of the node cannot generate this type of error.                                                                                                                                                                          | 0b0   |
| [3]   | UER  | Signaled or Recoverable Error generation. The value is:<br><br><b>0b0</b><br><br>The fault generation feature of the node cannot generate this type of error.                                                                                                                                                                        | 0b0   |
| [2]   | UEU  | Unrecoverable Error generation. The value is:<br><br><b>0b0</b><br><br>The fault generation feature of the node cannot generate this type of error.                                                                                                                                                                                  | 0b0   |
| [1]   | UC   | Uncontainable Error generation. The value is:<br><br><b>0b1</b><br><br>The fault generation feature of the node allows generation of this type of error.                                                                                                                                                                             | 0b1   |
| [0]   | OF   | Overflow flag. Describes how the fault generation feature of the node sets the ERXSTATUS_EL1.OF status flag. The value is:<br><br><b>0b0</b><br><br>When an injected error is recorded, the node sets ext-ERR<n>STATUS.OF according to the architecture-defined rules for setting the OF field. ext-ERR<n>PFGCTL.OF is <b>RES0</b> . | 0b0   |

## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXPFGF\_EL1 is RAZ.
- Direct reads of ERXPFGF\_EL1 are NOPs.
- Direct reads of ERXPFGF\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL selects an error record owned by a node that does not implement the Common Fault Injection Model Extension, then one of the following occurs:

- ERXPFGF\_EL1 is RAZ.
- Direct reads of ERXPFGF\_EL1 are NOPs.
- Direct reads of ERXPFGF\_EL1 are **UNDEFINED**.



A node does not implement the Common Fault Injection Model Extension if ERR<q>FR.INJ reads as 0b00. <q> is the index of the first error record owned by the same node as error record <n>, where <n> is the value in AArch64-ERRSELR\_EL1.SEL. If the node owns a single record then q = n.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>PFGF is not present, meaning reads of ERXPFGF\_EL1 are **RES0**.

ext-ERR<n>PFGF describes additional constraints that also apply when ext-ERR<n>PFGF is accessed through ERXPFGF\_EL1.

MRS <Xt>, ERXPFGF\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b100 |

### Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXPFGF\_EL1 is RAZ.
- Direct reads of ERXPFGF\_EL1 are NOPs.
- Direct reads of ERXPFGF\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL selects an error record owned by a node that does not implement the Common Fault Injection Model Extension, then one of the following occurs:

- ERXPFGF\_EL1 is RAZ.
- Direct reads of ERXPFGF\_EL1 are NOPs.
- Direct reads of ERXPFGF\_EL1 are UNDEFINED.



A node does not implement the Common Fault Injection Model Extension if ERR<q>FR.INJ reads as 0b00. <q> is the index of the first error record owned by the same node as error record <n>, where <n> is the value in AArch64-ERRSELR\_EL1.SEL. If the node owns a single record then q = n.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>PFGF is not present, meaning reads of ERXPFGF\_EL1 are RES0.

ext-ERR<n>PFGF describes additional constraints that also apply when ext-ERR<n>PFGF is accessed through ERXPFGF\_EL1.

MRS <Xt>, ERXPFGF\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.FIEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEN == '1' && HFGTR_EL2.ERXPFGF_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif SCR_EL3.FIEN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXPFGF_EL1;
    elsif PSTATE.EL == EL2 then
        if SCR_EL3.FIEN == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXPFGF_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = ERXPFGF_EL1;

```

## A.14.8 ERXPFGCTL\_EL1, Selected Pseudo-fault Generation Control Register

Accesses ext-ERR<n>PFGCTL for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

RAS registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 00xx | xxxx | xxxx | xxxx | xxx1 | x0x0 | 000x | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-139: AArch64\_erxpgctl\_el1 bit assignments**

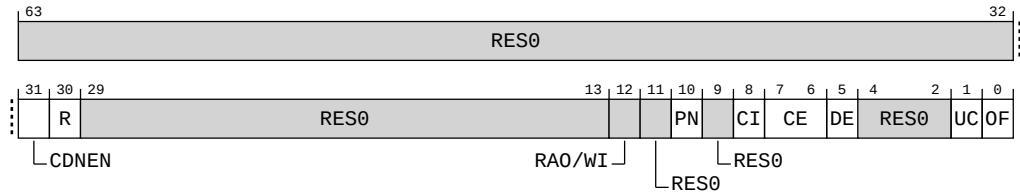


**Table A-380: ERXPFGCTL\_EL1 bit descriptions**

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Reset      |
|---------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [63:32] | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0       |
| [31]    | CDNEN  | Countdown Enable. Controls transfers from the value that is held in the ERXPFGCDN_EL1 into the Error Generation Counter and enables this counter.<br><br><b>0b0</b><br>The Error Generation Counter is disabled.<br><br><b>0b1</b><br>The Error Generation Counter is enabled. On a write of 0b1 to this bit, the Error Generation Counter is set to ERXPFGCDN_EL1.CDN.<br><br>Cold reset only. Unaffected by Warm reset | 0b0        |
| [30]    | R      | Restart. Controls whether, upon reaching zero, the Error Generation Counter restarts from the ERXPFGCDN_EL1 value or stops.<br><br><b>0b0</b><br>On reaching 0, the Error Generation Counter will stop.<br><br><b>0b1</b><br>On reaching 0, the Error Generation Counter is set to ERXPFGCDN_EL1.CDN.<br><br>Cold reset only. Unaffected by Warm reset                                                                   | 0b0        |
| [29:13] | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0       |
| [12]    | RAO/WI | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RAO/<br>WI |
| [11]    | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0       |
| [10]    | PN     | Poison flag. The value that is written to AArch64-ERXSTATUS.PN when an injected error is recorded.<br><br><b>0b00</b><br>AArch64-ERXSTATUS.PN is set to 0 when an injected error is recorded.<br><br><b>0b01</b><br>AArch64-ERXSTATUS.PN is set to 1 when an injected error is recorded.                                                                                                                                 | 0b0        |
| [9]     | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0       |

| Bits  | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [8]   | CI   | Critical Error flag. The value that is written to AArch64-ERXSTATUS.CI when an injected error is recorded.<br><br><b>0b0</b><br>AArch64-ERXSTATUS.CI is set to 0 when an injected error is recorded.<br><br><b>0b1</b><br>AArch64-ERXSTATUS.CI is set to 1 when an injected error is recorded.                                                                                                                                                              | 0b0   |
| [7:6] | CE   | Corrected Error generation enable. Controls the type of Corrected Error condition that might be generated. The possible values are:<br><br><b>0b00</b><br>No error of this type will be generated.<br><br><b>0b01</b><br>A non-specific Corrected Error, that is, a Corrected Error that is recorded as ERXSTATUS_EL1.CE == 0b10, might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset | 0b00  |
| [5]   | DE   | Deferred Error generation enable. The possible values are:<br><br><b>0b0</b><br>No error of this type will be generated.<br><br><b>0b1</b><br>An error of this type might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                               | 0b0   |
| [4:2] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [1]   | UC   | Uncontainable Error generation enable. The possible values are:<br><br><b>0b0</b><br>No error of this type will be generated.<br><br><b>0b1</b><br>An error of this type might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                          | 0b0   |
| [0]   | OF   | Uncontainable Error generation enable. The possible values are:<br><br><b>0b0</b><br>No error of this type will be generated.<br><br><b>0b1</b><br>An error of this type might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                          | 0b0   |

## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXPGCTL\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPGCTL\_EL1 are NOPs.

- Direct reads and writes of ERXPFGCTL\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL selects an error record owned by a node that does not implement the Common Fault Injection Model Extension, then one of the following occurs:

- ERXPFGCTL\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCTL\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCTL\_EL1 are **UNDEFINED**.



Note

A node does not implement the Common Fault Injection Model Extension if ERR<q>FR.INJ reads as 0b00. <q> is the index of the first error record owned by the same node as error record <n>, where <n> is the value in AArch64-ERRSELR\_EL1.SEL. If the node owns a single record then q = n.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>PFGCTL is not present, meaning reads and writes of ERXPFGCTL\_EL1 are **RES0**.

ext-ERR<n>PFGCTL describes additional constraints that also apply when ext-ERR<n>PFGCTL is accessed through ERXPFGCTL\_EL1.

MRS <Xt>, ERXPFGCTL\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b101 |

MSR ERXPFGCTL\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b101 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXPFGCTL\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCTL\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCTL\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL selects an error record owned by a node that does not implement the Common Fault Injection Model Extension, then one of the following occurs:

- ERXPFGCTL\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCTL\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCTL\_EL1 are UNDEFINED.



A node does not implement the Common Fault Injection Model Extension if ERR<q>FR.INJ reads as 0b00. <q> is the index of the first error record owned by the same node as error record <n>, where <n> is the value in AArch64-ERRSELR\_EL1.SEL. If the node owns a single record then q = n.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>PFGCTL is not present, meaning reads and writes of ERXPFGCTL\_EL1 are RESO.

ext-ERR<n>PFGCTL describes additional constraints that also apply when ext-ERR<n>PFGCTL is accessed through ERXPFGCTL\_EL1.

MRS <Xt>, ERXPFGCTL\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.FIEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXPFGCTL_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.FIEN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXPFGCTL_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.FIEN == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXPFGCTL_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERXPFGCTL_EL1;
```

MSR ERXPFGCTL\_EL1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.FIEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXPFGCTL_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.FIEN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXPFGCTL_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.FIEN == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                ERXPFGCTL_EL1 = X[t, 64];
```



```
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ERXPFPGCTL_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        ERXPFPGCTL_EL1 = X[t, 64];
```

### A.14.9 ERXPFPGCDN\_EL1, Selected Pseudo-fault Generation Countdown Register

Accesses ext-ERR<n>PFGCDN for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

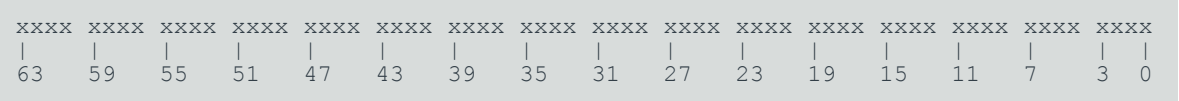
##### Functional group

RAS registers

##### Access type

See bit descriptions

##### Reset value



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure A-140: AArch64\_erxpfgcdn\_el1 bit assignments

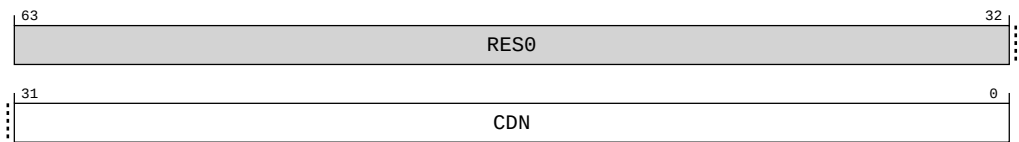


Table A-383: ERXPFPGCDN\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:32] | RES0 | Reserved    | RES0  |

| Bits   | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Reset  |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:0] | CDN  | <p>Countdown value.</p> <p>This field is copied to Error Generation Counter when either:</p> <ul style="list-style-type: none"> <li>Software writes ERXPFGCTL_EL1.CDNEN with 1.</li> <li>The Error Generation Counter decrements to zero and ERXPFGCTL_EL1.R == 0b1.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> <p><b>Note:</b><br/>The current Error Generation Counter value is not visible to software.</p> | 32 {x} |

### Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXPFGCDN\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCDN\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCDN\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL selects an error record owned by a node that does not implement the Common Fault Injection Model Extension, then one of the following occurs:

- ERXPFGCDN\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCDN\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCDN\_EL1 are **UNDEFINED**.



**Note**

A node does not implement the Common Fault Injection Model Extension if ERR<q>FR.INJ reads as 0b00. <q> is the index of the first error record owned by the same node as error record <n>, where <n> is the value in AArch64-ERRSELR\_EL1.SEL. If the node owns a single record then q = n.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>PFGCDN is not present, meaning reads and writes of ERXPFGCDN\_EL1 are **RESO**.

ext-ERR<n>PFGCDN describes additional constraints that also apply when ext-ERR<n>PFGCDN is accessed through ERXPFGCDN\_EL1.

MRS <Xt>, ERXPFGCDN\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b110 |

MSR ERXPFGCDN\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0100 | 0b110 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXPFGCDN\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCDN\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCDN\_EL1 are UNDEFINED.

If AArch64-ERRSELR\_EL1.SEL selects an error record owned by a node that does not implement the Common Fault Injection Model Extension, then one of the following occurs:

- ERXPFGCDN\_EL1 is RAZ/WI.
- Direct reads and writes of ERXPFGCDN\_EL1 are NOPs.
- Direct reads and writes of ERXPFGCDN\_EL1 are UNDEFINED.



Note

A node does not implement the Common Fault Injection Model Extension if ERR<q>FR.INJ reads as 0b00. <q> is the index of the first error record owned by the same node as error record <n>, where <n> is the value in AArch64-ERRSELR\_EL1.SEL. If the node owns a single record then q = n.

If AArch64-ERRSELR\_EL1.SEL is not the index of the first error record owned by a node, then ext-ERR<n>PFGCDN is not present, meaning reads and writes of ERXPFGCDN\_EL1 are RES0.

ext-ERR<n>PFGCDN describes additional constraints that also apply when ext-ERR<n>PFGCDN is accessed through ERXPFGCDN\_EL1.

MRS <Xt>, ERXPFGCDN\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.FIEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEN == '1' && HFGTR_EL2.ERXPFGCDN_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.FIEN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXPFGCDN_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.FIEN == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);

```

```

    else
        X[t, 64] = ERXPFGCDN_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = ERXPFGCDN_EL1;

```

MSR ERXPFGCDN\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.FIEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXPFGCDN_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif SCR_EL3.FIEN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXPFGCDN_EL1 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if SCR_EL3.FIEN == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ERXPFGCDN_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        ERXPFGCDN_EL1 = X[t, 64];

```

## A.14.10 ERXMISCO\_EL1, Selected Error Record Miscellaneous Register 0

Accesses ext-ERR<n>MISCO for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Functional group

RAS registers

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure A-141: AArch64\_erxmisc0\_el1 bit assignments**

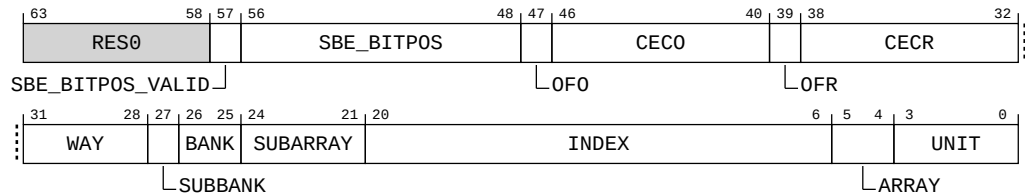


**Table A-386: ERXMISC0\_EL1 bit descriptions**

| Bits    | Name             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|---------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:58] | RES0             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [57]    | SBE_BITPOS_VALID | Single Bit Error (SBE) bit position field ERXMISC0_EL1.SBE_BITPOS contains valid data<br><br><b>0b0</b><br>ERXMISC0_EL1.SBE_BITPOS does not contain valid data.<br><br><b>0b1</b><br>ERXMISC0_EL1.SBE_BITPOS contains valid data.                                                                                                                                                                                                                                                           | x     |
| [56:48] | SBE_BITPOS       | Single Bit Error (SBE) bit position. For a correctable error in a RAM with ECC (L1 data cache, L2 cache), indicates the bit position of the corrected error. Valid when ERXMISC0_EL1.SBE_BITPOS_VALID is 1'b1                                                                                                                                                                                                                                                                               | 9 {x} |
| [47]    | OFO              | Sticky overflow bit, other. Set to 1 when ERXMISC0_EL1.CECO is incremented and wraps through zero.<br><br><b>0b0</b><br>Other counter has not overflowed.<br><br><b>0b1</b><br>Other counter has overflowed.<br><br>A direct write that modifies this bit might indirectly set ERXSTATUS_EL1.OF to an <b>UNKNOWN</b> value and a direct write to ERXSTATUS_EL1.OF that clears it to zero might indirectly set this bit to an <b>UNKNOWN</b> value.<br><br>Unaffected by Cold or Warm reset. | x     |
| [46:40] | CECO             | Corrected error count, other. Incremented for each countable error that is not accounted for by incrementing ERXMISC0_EL1.CECR.<br><br>Unaffected by Cold or Warm reset.                                                                                                                                                                                                                                                                                                                    | 7 {x} |

| Bits    | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|---------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [39]    | OFR     | <p>Sticky overflow bit, repeat. Set to 1 when ERXMISCO_EL1.CECR is incremented and wraps through zero.</p> <p><b>0b0</b><br/>Repeat counter has not overflowed.</p> <p><b>0b1</b><br/>Repeat counter has overflowed.</p> <p>A direct write that modifies this bit might indirectly set ERXSTATUS_EL1.OF to an <b>UNKNOWN</b> value and a direct write to ERXSTATUS_EL1.OF that clears it to zero might indirectly set this bit to an <b>UNKNOWN</b> value.</p> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                   | x     |
| [38:32] | CECR    | <p>Corrected error count, repeat. Incremented for the first countable error, which also records other syndrome for the error, and subsequently for each countable error that matches the recorded other syndrome.</p> <p>This field resets to an IMPLEMENTATION DEFINED which might be <b>UNKNOWN</b> on a Cold reset. If the reset value is <b>UNKNOWN</b>, then the value of this field remains <b>UNKNOWN</b> until software initializes it.</p> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                              | 7 {x} |
| [31:28] | WAY     | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L1 Data Cache]</p> <ul style="list-style-type: none"> <li>Indicates which Tag RAM way or data RAM way detected the error. Upper 2 bits are unused.</li> </ul> <p>[L2 TLB]</p> <ul style="list-style-type: none"> <li>Indicates which RAM detected an error. The possible values are 0 (RAM 1) to 9 (RAM 10).</li> </ul> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which way detected the error. Upper 2 bits are unused.</li> </ul> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which way detected the error.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> | xxxx  |
| [27]    | SUBBANK | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which subbank detected the error, valid for Instruction Data Cache. For Tag errors this field is zero.</li> </ul> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                                                                                                                                                         | x     |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [26:25] | BANK     | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which L2 bank detected the error.</li> </ul> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which bank detected the error, valid for Instruction Data Cache. For Tag errors this field is zero.</li> </ul> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                                                                                                                          | xx    |
| [24:21] | SUBARRAY | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which L2 data ECC granule detected the error. Upper bits are unused dependent on the ECC granule size.</li> </ul> <p>[L1 Data Cache]</p> <ul style="list-style-type: none"> <li>Indicates for L1 Data RAM which word had the error detected. For L1 Tag RAMs which bank had the error (0b0000: bank0, 0b0001: bank1)</li> </ul> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                                     | xxxx  |
| [20:6]  | INDEX    | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which index detected the error. Upper bits of the index are unused depending on the cache size.</li> </ul> <p>[L1 Data Cache]</p> <ul style="list-style-type: none"> <li>Indicates which index detected the error. Upper bits of the index are unused depending on the cache size</li> </ul> <p>[L2 TLB]</p> <ul style="list-style-type: none"> <li>Index of TLB RAM. Upper 4 bits are unused.</li> </ul> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which index detected the error. Upper bits of the index are unused depending on the cache size.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> | 15{x} |

| Bits  | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|-------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [5:4] | ARRAY | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <p>Indicates which array detected the error. The possible values are:</p> <ul style="list-style-type: none"> <li>0b00 L2 Tag RAM.</li> <li>0b01 L2 Data RAM.</li> <li>0b10 L2 TQ Data RAM.</li> <li>0b11 CHI Error.</li> </ul> <p>[L1 Data Cache]</p> <p>Indicates which array detected the error. The possible values are:</p> <ul style="list-style-type: none"> <li>00 LS Tag RAM 0.</li> <li>01 LS Tag RAM 1.</li> <li>10 LS Data RAM.</li> <li>11 LS Tag RAM 2.</li> </ul> <p>[L2 TLB]</p> <p>Indicates which array detected the error. The possible values are:</p> <ul style="list-style-type: none"> <li>00 Translation cache.</li> <li>01 GPT cache (when LEGACY_TZ_EN is 0).</li> <li>10 Reserved.</li> <li>11 Reserved.</li> </ul> <p>[L1 Instruction Cache]</p> <p>Indicates which array that detected the error, Data Array has higher priority. The possible values are:</p> <ul style="list-style-type: none"> <li>0b00 Tag.</li> <li>0b01 Data.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> | xx    |
| [3:0] | UNIT  | <p>Indicates the unit which detected the error. The possible values are:</p> <p><b>0b0001</b><br/>L1 Instruction Cache.</p> <p><b>0b0010</b><br/>L2 TLB.</p> <p><b>0b0100</b><br/>L1 Data Cache.</p> <p><b>0b1000</b><br/>L2 Cache.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | xxxx  |



## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXMISCO\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISCO\_EL1 are NOPs.
- Direct reads and writes of ERXMISCO\_EL1 are UNDEFINED.

ext-ERR<n>MISCO describes additional constraints that also apply when ext-ERR<n>MISCO is accessed through ERXMISCO\_EL1.

MRS <Xt>, ERXMISCO\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b000 |

MSR ERXMISCO\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b000 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXMISCO\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISCO\_EL1 are NOPs.
- Direct reads and writes of ERXMISCO\_EL1 are UNDEFINED.

ext-ERR<n>MISCO describes additional constraints that also apply when ext-ERR<n>MISCO is accessed through ERXMISCO\_EL1.

MRS <Xt>, ERXMISCO\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXMISCN_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXMISCO_EL1;
    elsif PSTATE.EL == EL2 then

```

```

if SCR_EL3.TERR == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ERXMISCO_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ERXMISCO_EL1;

```

MSR ERXMISCO\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXMISCO_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ERXMISCO_EL1 = X[t, 64];
    elseif PSTATE.EL == EL3 then
        ERXMISCO_EL1 = X[t, 64];

```

### A.14.11 ERXMISC1\_EL1, Selected Error Record Miscellaneous Register 1

Accesses ext-ERR<n>MISC1 for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

RAS registers

##### Access type

See bit descriptions

##### Reset value

```

xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

```



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-142: AArch64\_erxmisc1\_el1 bit assignments

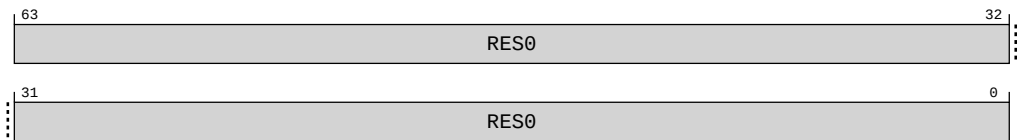


Table A-389: ERXMISC1\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

**Access**

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXMISC1\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISC1\_EL1 are NOPs.
- Direct reads and writes of ERXMISC1\_EL1 are UNDEFINED.

ext-ERR<n>MISC1 describes additional constraints that also apply when ext-ERR<n>MISC1 is accessed through ERXMISC1\_EL1.

MRS <Xt>, ERXMISC1\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b001 |

MSR ERXMISC1\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b001 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXMISC1\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISC1\_EL1 are NOPs.
- Direct reads and writes of ERXMISC1\_EL1 are UNDEFINED.

ext-ERR<n>MISC1 describes additional constraints that also apply when ext-ERR<n>MISC1 is accessed through ERXMISC1\_EL1.

MRS <Xt>, ERXMISC1\_EL1

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ERXMISC1_EL1;
elseif PSTATE.EL == EL2 then
    if SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ERXMISC1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ERXMISC1_EL1;
```

MSR ERXMISC1\_EL1, <Xt>

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ERXMISC1_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
    if SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
```

```
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXMISC1_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        ERXMISC1_EL1 = X[t, 64];
```

A.14.12 ERXMISC2\_EL1, Selected Error Record Miscellaneous Register 2

Accesses ext-ERR<n>MISC2 for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

Configurations

This register is available in all configurations.

Attributes

Width

64

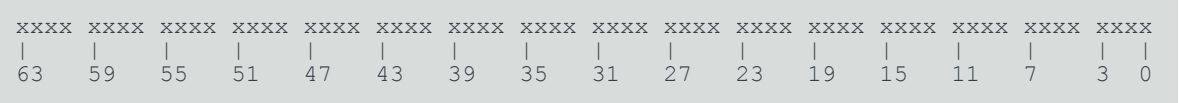
Functional group

RAS registers

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-143: AArch64\_erxmisc2\_el1 bit assignments

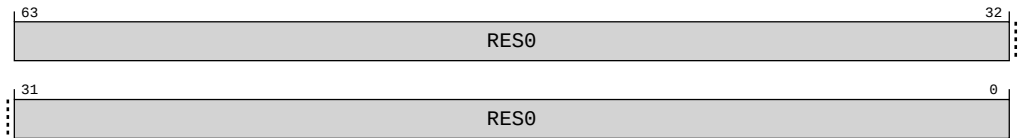


Table A-392: ERXMISC2\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

## Access

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXMISC2\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISC2\_EL1 are NOPs.
- Direct reads and writes of ERXMISC2\_EL1 are UNDEFINED.

ext-ERR<n>MISC2 describes additional constraints that also apply when ext-ERR<n>MISC2 is accessed through ERXMISC2\_EL1.

MRS <Xt>, ERXMISC2\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b010 |

MSR ERXMISC2\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b010 |

## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXMISC2\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISC2\_EL1 are NOPs.
- Direct reads and writes of ERXMISC2\_EL1 are UNDEFINED.

ext-ERR<n>MISC2 describes additional constraints that also apply when ext-ERR<n>MISC2 is accessed through ERXMISC2\_EL1.

MRS <Xt>, ERXMISC2\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXMISC2_EL1;
    elsif PSTATE.EL == EL2 then

```

```

if SCR_EL3.TERR == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ERXMISC2_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ERXMISC2_EL1;

```

MSR ERXMISC2\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXMISC2_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ERXMISC2_EL1 = X[t, 64];
    elseif PSTATE.EL == EL3 then
        ERXMISC2_EL1 = X[t, 64];

```

### A.14.13 ERXMISC3\_EL1, Selected Error Record Miscellaneous Register 3

Accesses ext-ERR<n>MISC3 for the error record <n> selected by AArch64-ERRSELR\_EL1.SEL.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

RAS registers

##### Access type

See bit descriptions

##### Reset value

```

xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

```



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-144: AArch64\_erxmisc3\_el1 bit assignments

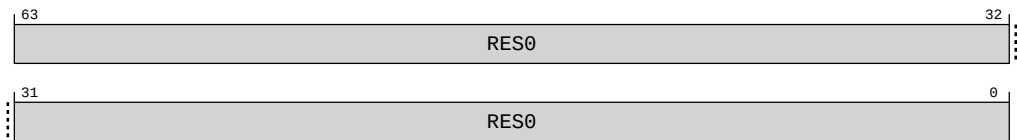


Table A-395: ERXMISC3\_EL1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

**Access**

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An **UNKNOWN** error record is selected.
- ERXMISC3\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISC3\_EL1 are NOPs.
- Direct reads and writes of ERXMISC3\_EL1 are UNDEFINED.

ext-ERR<n>MISC3 describes additional constraints that also apply when ext-ERR<n>MISC3 is accessed through ERXMISC3\_EL1.

MRS <Xt>, ERXMISC3\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b011 |

MSR ERXMISC3\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b0101 | 0b0101 | 0b011 |



## Accessibility

If AArch64-ERRIDR\_EL1.NUM is 0x0000 or AArch64-ERRSELR\_EL1.SEL is greater than or equal to AArch64-ERRIDR\_EL1.NUM, then one of the following occurs:

- An UNKNOWN error record is selected.
- ERXMISC3\_EL1 is RAZ/WI.
- Direct reads and writes of ERXMISC3\_EL1 are NOPs.
- Direct reads and writes of ERXMISC3\_EL1 are UNDEFINED.

ext-ERR<n>MISC3 describes additional constraints that also apply when ext-ERR<n>MISC3 is accessed through ERXMISC3\_EL1.

MRS <Xt>, ERXMISC3\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if HalTED() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ERXMISC3_EL1;
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if HalTED() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ERXMISC3_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = ERXMISC3_EL1;

```

MSR ERXMISC3\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TERR == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && SCR_EL3.FGTEn == '1' && HFGWTR_EL2.ERXMISCn_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif SCR_EL3.TERR == '1' then
        if HalTED() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            ERXMISC3_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if SCR_EL3.TERR == '1' then
            if HalTED() && EDSCR.SDD == '1' then
                UNDEFINED;

```

```

else
    AArch64.SystemAccessTrap(EL3, 0x18);
else
    ERXMISC3_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    ERXMISC3_EL1 = X[t, 64];

```

## A.15 AArch64 Statistical Profiling Extension registers summary

The following summary table provides an overview of all Statistical Profiling Extension registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-398: Statistical Profiling Extension registers summary**

| Name                         | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                                  |
|------------------------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------------|
| PMSCR_EL1                    | 3   | 0   | C9  | C9  | 0   | See individual bit resets. | 64-bit | Statistical Profiling Control Register (EL1) |
| <a href="#">PMSNEVFR_EL1</a> | 3   | 0   | C9  | C9  | 1   | See individual bit resets. | 64-bit | Sampling Inverted Event Filter Register      |
| PMSICR_EL1                   | 3   | 0   | C9  | C9  | 2   | See individual bit resets. | 64-bit | Sampling Interval Counter Register           |
| PMSIRR_EL1                   | 3   | 0   | C9  | C9  | 3   | See individual bit resets. | 64-bit | Sampling Interval Reload Register            |
| PMSFCR_EL1                   | 3   | 0   | C9  | C9  | 4   | See individual bit resets. | 64-bit | Sampling Filter Control Register             |
| <a href="#">PMSEVFR_EL1</a>  | 3   | 0   | C9  | C9  | 5   | See individual bit resets. | 64-bit | Sampling Event Filter Register               |
| PMSLATFR_EL1                 | 3   | 0   | C9  | C9  | 6   | See individual bit resets. | 64-bit | Sampling Latency Filter Register             |
| <a href="#">PMSIDR_EL1</a>   | 3   | 0   | C9  | C9  | 7   | See individual bit resets. | 64-bit | Sampling Profiling ID Register               |
| PMBLIMITR_EL1                | 3   | 0   | C9  | C10 | 0   | See individual bit resets. | 64-bit | Profiling Buffer Limit Address Register      |
| PMBPTR_EL1                   | 3   | 0   | C9  | C10 | 1   | See individual bit resets. | 64-bit | Profiling Buffer Write Pointer Register      |
| PMBSR_EL1                    | 3   | 0   | C9  | C10 | 3   | See individual bit resets. | 64-bit | Profiling Buffer Status/syndrome Register    |
| <a href="#">PMBIDR_EL1</a>   | 3   | 0   | C9  | C10 | 7   | See individual bit resets. | 64-bit | Profiling Buffer ID Register                 |
| PMSCR_EL2                    | 3   | 4   | C9  | C9  | 0   | See individual bit resets. | 64-bit | Statistical Profiling Control Register (EL2) |

### A.15.1 PMSNEVFR\_EL1, Sampling Inverted Event Filter Register

Controls sample filtering by events. The overall inverted filter is the logical OR of these filters. For example, if PMSNEVFR\_EL1.E[3] and PMSNEVFR\_EL1.E[5] are both set to 1, samples that have either event 3 (Level 1 unified or data cache refill) or event 5 (TLB walk) set to 1 are not recorded.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Statistical Profiling Extension registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0xx0 | 0000 | xxxx | xxxx | xxx0 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Note

Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure A-145: AArch64\_pmsnevfr\_el1 bit assignments

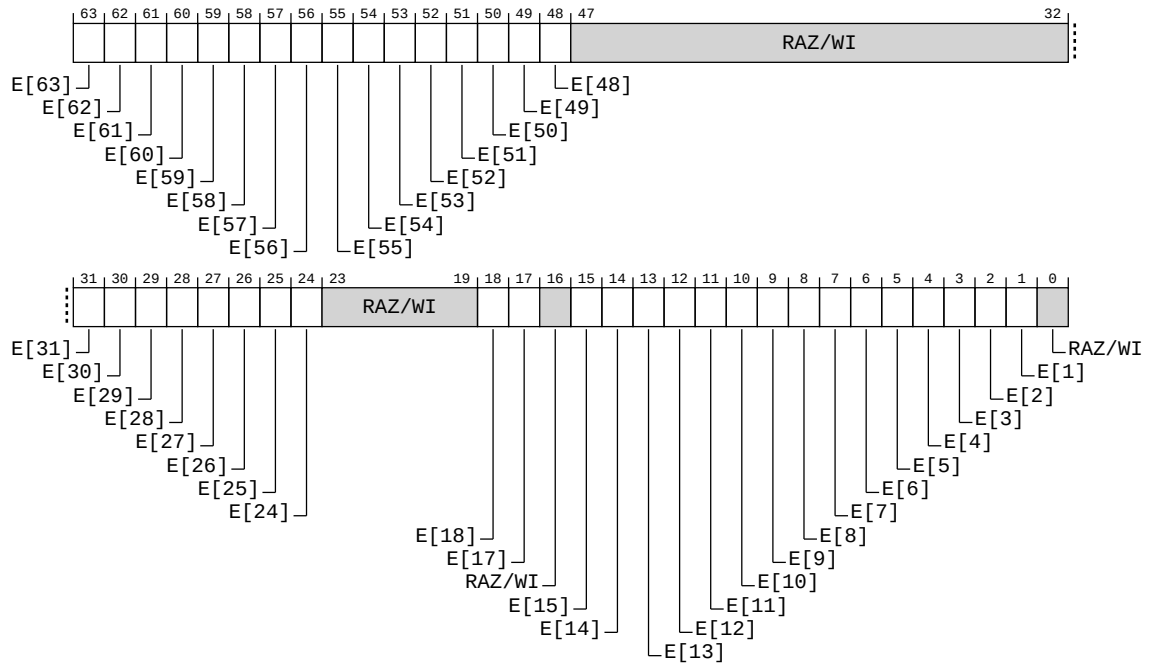


Table A-399: PMSNEVFR\_EL1 bit descriptions

| Bits | Name  | Description                                                                                                                                                                                                                                      | Reset |
|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63] | E[63] | <p>E[63] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 63.</p> <p><b>0b0</b></p> <p>Event 63 is ignored.</p> <p><b>0b1</b></p> <p>Do not record samples that have event 63 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [62] | E[62] | <p>E[62] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 62.</p> <p><b>0b0</b></p> <p>Event 62 is ignored.</p> <p><b>0b1</b></p> <p>Do not record samples that have event 62 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |

| Bits | Name  | Description                                                                                                                                                                                                                                | Reset |
|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [61] | E[61] | <p>E[61] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 61.</p> <p><b>0b0</b><br/>Event 61 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 61 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [60] | E[60] | <p>E[60] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 60.</p> <p><b>0b0</b><br/>Event 60 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 60 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [59] | E[59] | <p>E[59] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 59.</p> <p><b>0b0</b><br/>Event 59 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 59 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [58] | E[58] | <p>E[58] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 58.</p> <p><b>0b0</b><br/>Event 58 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 58 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [57] | E[57] | <p>E[57] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 57.</p> <p><b>0b0</b><br/>Event 57 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 57 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [56] | E[56] | <p>E[56] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 56.</p> <p><b>0b0</b><br/>Event 56 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 56 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |

| Bits | Name  | Description                                                                                                                                                                                                                                | Reset |
|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [55] | E[55] | <p>E[55] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 55.</p> <p><b>0b0</b><br/>Event 55 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 55 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [54] | E[54] | <p>E[54] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 54.</p> <p><b>0b0</b><br/>Event 54 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 54 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [53] | E[53] | <p>E[53] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 53.</p> <p><b>0b0</b><br/>Event 53 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 53 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [52] | E[52] | <p>E[52] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 52.</p> <p><b>0b0</b><br/>Event 52 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 52 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [51] | E[51] | <p>E[51] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 51.</p> <p><b>0b0</b><br/>Event 51 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 51 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |
| [50] | E[50] | <p>E[50] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 50.</p> <p><b>0b0</b><br/>Event 50 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 50 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0   |

| Bits    | Name          | Description                                                                                                                                                                                                                                | Reset         |
|---------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [49]    | E[49]         | <p>E[49] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 49.</p> <p><b>0b0</b><br/>Event 49 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 49 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [48]    | E[48]         | <p>E[48] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 48.</p> <p><b>0b0</b><br/>Event 48 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 48 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [47:32] | <b>RAZ/WI</b> | Reserved                                                                                                                                                                                                                                   | <b>RAZ/WI</b> |
| [31]    | E[31]         | <p>E[31] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 31.</p> <p><b>0b0</b><br/>Event 31 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 31 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [30]    | E[30]         | <p>E[30] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 30.</p> <p><b>0b0</b><br/>Event 30 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 30 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [29]    | E[29]         | <p>E[29] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 29.</p> <p><b>0b0</b><br/>Event 29 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 29 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [28]    | E[28]         | <p>E[28] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 28.</p> <p><b>0b0</b><br/>Event 28 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 28 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |

| Bits    | Name          | Description                                                                                                                                                                                                                                | Reset         |
|---------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [27]    | E[27]         | <p>E[27] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 27.</p> <p><b>0b0</b><br/>Event 27 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 27 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [26]    | E[26]         | <p>E[26] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 26.</p> <p><b>0b0</b><br/>Event 26 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 26 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [25]    | E[25]         | <p>E[25] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 25.</p> <p><b>0b0</b><br/>Event 25 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 25 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [24]    | E[24]         | <p>E[24] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 24.</p> <p><b>0b0</b><br/>Event 24 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 24 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p> | 0b0           |
| [23:19] | <b>RAZ/WI</b> | Reserved                                                                                                                                                                                                                                   | <b>RAZ/WI</b> |
| [18]    | E[18]         | <p>Not empty predicate.</p> <p><b>0b0</b><br/>Empty predicate event is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have the Empty predicate event == 1.</p>                                                                  | x             |
| [17]    | E[17]         | <p>Not partial predicate.</p> <p><b>0b0</b><br/>Partial predicate event is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have the Partial predicate event == 1.</p>                                                            | x             |
| [16]    | <b>RAZ/WI</b> | Reserved                                                                                                                                                                                                                                   | <b>RAZ/WI</b> |



| Bits | Name  | Description                                                                                                                                                                                                                                                                                            | Reset |
|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [15] | E[15] | <p>E[15] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 15.</p> <p><b>0b0</b><br/>Event 15 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 15 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p>                                                             | 0b0   |
| [14] | E[14] | <p>E[14] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 14.</p> <p><b>0b0</b><br/>Event 14 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 14 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p>                                                             | 0b0   |
| [13] | E[13] | <p>E[13] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 13.</p> <p><b>0b0</b><br/>Event 13 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 13 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p>                                                             | 0b0   |
| [12] | E[12] | <p>E[12] is the event filter for <b>IMPLEMENTATION DEFINED</b> event 12.</p> <p><b>0b0</b><br/>Event 12 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 12 == 1.</p> <p>Access to this field is: <b>RAZ/WI</b></p>                                                             | 0b0   |
| [11] | E[11] | <p>Aligned.</p> <p><b>0b0</b><br/>Misalignment event is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have the Misalignment event == 1.</p>                                                                                                                                                | x     |
| [10] | E[10] | <p><b>When filtering on event 10 is optionally supported &amp;&amp; event 10 is implemented</b><br/>No remote access.</p> <p><b>0b0</b><br/>Remote access event is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have the Remote access event == 1.</p> <p><b>Otherwise</b><br/>RAZ/WI</p> | xxxx  |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [9]  | E[9] | <b>When filtering on event 9 is optionally supported &amp;&amp; event 9 is implemented</b><br>Last Level cache hit.<br><b>0b0</b><br>Last Level cache miss event is ignored.<br><b>0b1</b><br>Do not record samples that have the Last Level cache miss event == 1.<br><br><b>Otherwise</b><br>RAZ/WI           | xxxx  |
| [8]  | E[8] | <b>When filtering on event 8 is optionally supported &amp;&amp; event 8 is implemented</b><br>No Last Level cache access.<br><b>0b0</b><br>Last Level cache access event is ignored.<br><b>0b1</b><br>Do not record samples that have the Last Level cache access event == 1.<br><br><b>Otherwise</b><br>RAZ/WI | xxxx  |
| [7]  | E[7] | Correctly predicted.<br><b>0b0</b><br>Mispredicted event is ignored.<br><b>0b1</b><br>Do not record samples that have the Mispredicted event == 1.                                                                                                                                                              | x     |
| [6]  | E[6] | Taken.<br><b>0b0</b><br>Not taken event is ignored.<br><b>0b1</b><br>Do not record samples that have the Not taken event == 1.                                                                                                                                                                                  | x     |
| [5]  | E[5] | TLB hit.<br><b>0b0</b><br>TLB walk event is ignored.<br><b>0b1</b><br>Do not record samples that have the TLB walk event == 1.                                                                                                                                                                                  | x     |
| [4]  | E[4] | <b>When filtering on event 4 is optionally supported</b><br>No TLB access.<br><b>0b0</b><br>TLB access event is ignored.<br><b>0b1</b><br>Do not record samples that have the TLB access event == 1.<br><br><b>Otherwise</b><br>RAZ/WI                                                                          | xxxx  |

| Bits | Name   | Description                                                                                                                                                                                                                                                                                 | Reset  |
|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [3]  | E[3]   | Level 1 data or unified cache hit.<br><br><b>0b0</b><br>Level 1 data or unified cache refill event is ignored.<br><br><b>0b1</b><br>Do not record samples that have the Level 1 data or unified cache refill event == 1.                                                                    | x      |
| [2]  | E[2]   | <b>When filtering on event 2 is optionally supported</b><br>No Level 1 data cache access.<br><br><b>0b0</b><br>Level 1 data cache access event is ignored.<br><br><b>0b1</b><br>Do not record samples that have the Level 1 data cache access event == 1.<br><br><b>Otherwise</b><br>RAZ/WI | xxxx   |
| [1]  | E[1]   | Speculative.<br><br><b>0b0</b><br>Architecturally executed event is ignored.<br><br><b>0b1</b><br>Do not record samples that have the Architecturally executed event == 1.                                                                                                                  | x      |
| [0]  | RAZ/WI | Reserved                                                                                                                                                                                                                                                                                    | RAZ/WI |

## Access

MRS <Xt>, PMSNEVFR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b001 |

MSR PMSNEVFR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b001 |

## Accessibility

MRS <Xt>, PMSNEVFR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.nPMSNEVFR_EL1 == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPMS == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        elsif MDCR_EL3.EnPMSN == '0' then

```

```

        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMSNEVFR_EL1;
    elseif PSTATE.EL == EL2 then
        if MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            elseif MDCR_EL3.EnPMSN == '0' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMSNEVFR_EL1;
    elseif PSTATE.EL == EL3 then
        X[t, 64] = PMSNEVFR_EL1;

```

## MSR PMSNEVFR\_EL1, &lt;Xt&gt;

```

    if PSTATE.EL == EL0 then
        UNDEFINED;
    elseif PSTATE.EL == EL1 then
        if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGWTR_EL2.nPMSNEVFR_EL1 == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif EL2Enabled() && MDCR_EL2.TPMS == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            elseif MDCR_EL3.EnPMSN == '0' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                PMSNEVFR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL2 then
        if MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            elseif MDCR_EL3.EnPMSN == '0' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            else
                PMSNEVFR_EL1 = X[t, 64];
    elseif PSTATE.EL == EL3 then
        PMSNEVFR_EL1 = X[t, 64];

```

### A.15.2 PMSEVFR\_EL1, Sampling Event Filter Register

Controls sample filtering by events. The overall filter is the logical AND of these filters. For example, if PMSEVFR\_EL1.E[3] and PMSEVFR\_EL1.E[5] are both set to 1, only samples that have both event 3 (Level 1 unified or data cache refill) and event 5 (TLB walk) set to 1 are recorded.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Statistical Profiling Extension registers

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 | xxxx | xxxx | 0000 | 0xx0 | xxxx | xxxx | xxxx | xxx0 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure A-146: AArch64\_pmsevfr\_el1 bit assignments

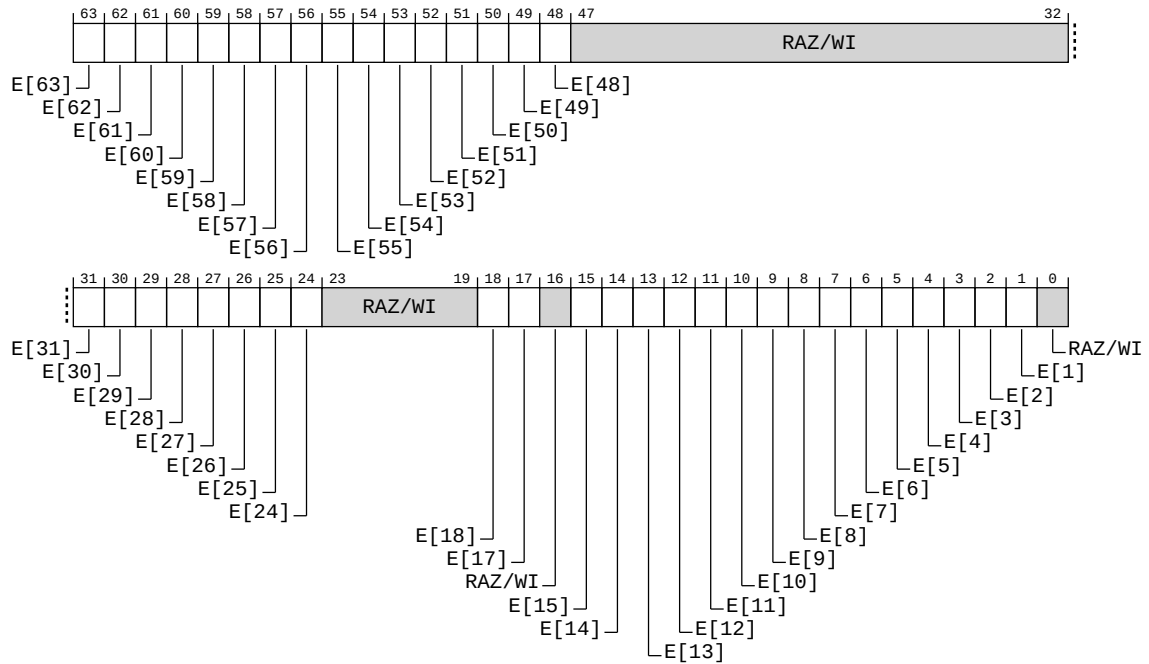


Table A-402: PMSEVFR\_EL1 bit descriptions

| Bits | Name  | Description                                                                                                                                                                                                                                                                             | Reset |
|------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63] | E[63] | <p>E[63] is the event filter for event 63. If event 63 is not implemented, or filtering on event 63 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b></p> <p>Event 63 is ignored.</p> <p><b>0b1</b></p> <p>Do not record samples that have event 63 == 0.</p> | x     |
| [62] | E[62] | <p>E[62] is the event filter for event 62. If event 62 is not implemented, or filtering on event 62 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b></p> <p>Event 62 is ignored.</p> <p><b>0b1</b></p> <p>Do not record samples that have event 62 == 0.</p> | x     |
| [61] | E[61] | <p>E[61] is the event filter for event 61. If event 61 is not implemented, or filtering on event 61 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b></p> <p>Event 61 is ignored.</p> <p><b>0b1</b></p> <p>Do not record samples that have event 61 == 0.</p> | x     |

| Bits | Name  | Description                                                                                                                                                                                                                                                                       | Reset |
|------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [60] | E[60] | <p>E[60] is the event filter for event 60. If event 60 is not implemented, or filtering on event 60 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 60 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 60 == 0.</p> | x     |
| [59] | E[59] | <p>E[59] is the event filter for event 59. If event 59 is not implemented, or filtering on event 59 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 59 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 59 == 0.</p> | x     |
| [58] | E[58] | <p>E[58] is the event filter for event 58. If event 58 is not implemented, or filtering on event 58 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 58 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 58 == 0.</p> | x     |
| [57] | E[57] | <p>E[57] is the event filter for event 57. If event 57 is not implemented, or filtering on event 57 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 57 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 57 == 0.</p> | x     |
| [56] | E[56] | <p>E[56] is the event filter for event 56. If event 56 is not implemented, or filtering on event 56 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 56 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 56 == 0.</p> | x     |
| [55] | E[55] | <p>E[55] is the event filter for event 55. If event 55 is not implemented, or filtering on event 55 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 55 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 55 == 0.</p> | x     |
| [54] | E[54] | <p>E[54] is the event filter for event 54. If event 54 is not implemented, or filtering on event 54 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 54 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 54 == 0.</p> | x     |

| Bits    | Name          | Description                                                                                                                                                                                                                                                                       | Reset         |
|---------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [53]    | E[53]         | <p>E[53] is the event filter for event 53. If event 53 is not implemented, or filtering on event 53 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 53 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 53 == 0.</p> | x             |
| [52]    | E[52]         | <p>E[52] is the event filter for event 52. If event 52 is not implemented, or filtering on event 52 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 52 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 52 == 0.</p> | x             |
| [51]    | E[51]         | <p>E[51] is the event filter for event 51. If event 51 is not implemented, or filtering on event 51 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 51 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 51 == 0.</p> | x             |
| [50]    | E[50]         | <p>E[50] is the event filter for event 50. If event 50 is not implemented, or filtering on event 50 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 50 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 50 == 0.</p> | x             |
| [49]    | E[49]         | <p>E[49] is the event filter for event 49. If event 49 is not implemented, or filtering on event 49 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 49 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 49 == 0.</p> | x             |
| [48]    | E[48]         | <p>E[48] is the event filter for event 48. If event 48 is not implemented, or filtering on event 48 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 48 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 48 == 0.</p> | x             |
| [47:32] | <b>RAZ/WI</b> | Reserved                                                                                                                                                                                                                                                                          | <b>RAZ/WI</b> |
| [31]    | E[31]         | <p>E[31] is the event filter for event 31. If event 31 is not implemented, or filtering on event 31 is not supported, the corresponding bit is <b>RAZ/WI</b>.</p> <p><b>0b0</b><br/>Event 31 is ignored.</p> <p><b>0b1</b><br/>Do not record samples that have event 31 == 0.</p> | x             |



| Bits    | Name          | Description                                                                                                                                                                                                                                                               | Reset         |
|---------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [30]    | E[30]         | E[30] is the event filter for event 30. If event 30 is not implemented, or filtering on event 30 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 30 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 30 == 0. | x             |
| [29]    | E[29]         | E[29] is the event filter for event 29. If event 29 is not implemented, or filtering on event 29 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 29 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 29 == 0. | x             |
| [28]    | E[28]         | E[28] is the event filter for event 28. If event 28 is not implemented, or filtering on event 28 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 28 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 28 == 0. | x             |
| [27]    | E[27]         | E[27] is the event filter for event 27. If event 27 is not implemented, or filtering on event 27 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 27 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 27 == 0. | x             |
| [26]    | E[26]         | E[26] is the event filter for event 26. If event 26 is not implemented, or filtering on event 26 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 26 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 26 == 0. | x             |
| [25]    | E[25]         | E[25] is the event filter for event 25. If event 25 is not implemented, or filtering on event 25 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 25 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 25 == 0. | x             |
| [24]    | E[24]         | E[24] is the event filter for event 24. If event 24 is not implemented, or filtering on event 24 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 24 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 24 == 0. | x             |
| [23:19] | <b>RAZ/WI</b> | Reserved                                                                                                                                                                                                                                                                  | <b>RAZ/WI</b> |

| Bits | Name                     | Description                                                                                                                                                                                                                                                               | Reset                    |
|------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| [18] | E[18]                    | Empty predicate.<br><br><b>0b0</b><br>Empty predicate event is ignored.<br><br><b>0b1</b><br>Do not record samples that have the Empty predicate event == 0.                                                                                                              | x                        |
| [17] | E[17]                    | Partial predicate.<br><br><b>0b0</b><br>Partial predicate event is ignored.<br><br><b>0b1</b><br>Do not record samples that have the Partial predicate event == 0.                                                                                                        | x                        |
| [16] | <b>RAZ/</b><br><b>WI</b> | Reserved                                                                                                                                                                                                                                                                  | <b>RAZ/</b><br><b>WI</b> |
| [15] | E[15]                    | E[15] is the event filter for event 15. If event 15 is not implemented, or filtering on event 15 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 15 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 15 == 0. | x                        |
| [14] | E[14]                    | E[14] is the event filter for event 14. If event 14 is not implemented, or filtering on event 14 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 14 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 14 == 0. | x                        |
| [13] | E[13]                    | E[13] is the event filter for event 13. If event 13 is not implemented, or filtering on event 13 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 13 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 13 == 0. | x                        |
| [12] | E[12]                    | E[12] is the event filter for event 12. If event 12 is not implemented, or filtering on event 12 is not supported, the corresponding bit is <b>RAZ/WI</b> .<br><br><b>0b0</b><br>Event 12 is ignored.<br><br><b>0b1</b><br>Do not record samples that have event 12 == 0. | x                        |
| [11] | E[11]                    | Alignment.<br><br><b>0b0</b><br>Alignment event is ignored.<br><br><b>0b1</b><br>Do not record samples that have the Alignment event == 0.                                                                                                                                | x                        |

| Bits | Name  | Description                                                                                                                                                                                                                                                                                                  | Reset |
|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [10] | E[10] | <b>When filtering on event 10 is optionally supported &amp;&amp; event 10 is implemented</b><br>Remote access.<br><b>0b0</b><br>Remote access event is ignored.<br><b>0b1</b><br>Do not record samples that have the Remote access event == 0.<br><br><b>Otherwise</b><br>RAZ/WI                             | xxxx  |
| [9]  | E[9]  | <b>When filtering on event 9 is optionally supported &amp;&amp; event 9 is implemented</b><br>Last Level cache miss.<br><b>0b0</b><br>Last Level cache miss event is ignored.<br><b>0b1</b><br>Do not record samples that have the Last Level cache miss event == 0.<br><br><b>Otherwise</b><br>RAZ/WI       | xxxx  |
| [8]  | E[8]  | <b>When filtering on event 8 is optionally supported &amp;&amp; event 8 is implemented</b><br>Last Level cache access.<br><b>0b0</b><br>Last Level cache access event is ignored.<br><b>0b1</b><br>Do not record samples that have the Last Level cache access event == 0.<br><br><b>Otherwise</b><br>RAZ/WI | xxxx  |
| [7]  | E[7]  | Mispredicted.<br><b>0b0</b><br>Mispredicted event is ignored.<br><b>0b1</b><br>Do not record samples that have the Mispredicted event == 0.                                                                                                                                                                  | x     |
| [6]  | E[6]  | Not taken.<br><b>0b0</b><br>Not taken event is ignored.<br><b>0b1</b><br>Do not record samples that have the Not taken event == 0.                                                                                                                                                                           | x     |
| [5]  | E[5]  | TLB walk.<br><b>0b0</b><br>TLB walk event is ignored.<br><b>0b1</b><br>Do not record samples that have the TLB walk event == 0.                                                                                                                                                                              | x     |

| Bits | Name       | Description                                                                                                                                                                                                                                                                      | Reset      |
|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [4]  | E[4]       | <b>When filtering on event 4 is optionally supported</b><br>TLB access.<br><b>0b0</b><br>TLB access event is ignored.<br><b>0b1</b><br>Do not record samples that have the TLB access event == 0.<br><br><b>Otherwise</b><br>RAZ/WI                                              | xxxx       |
| [3]  | E[3]       | Level 1 data or unified cache refill.<br><b>0b0</b><br>Level 1 data or unified cache refill event is ignored.<br><b>0b1</b><br>Do not record samples that have the Level 1 data or unified cache refill event == 0.                                                              | x          |
| [2]  | E[2]       | <b>When filtering on event 2 is optionally supported</b><br>Level 1 data cache access.<br><b>0b0</b><br>Level 1 data cache access event is ignored.<br><b>0b1</b><br>Do not record samples that have the Level 1 data cache access event == 0.<br><br><b>Otherwise</b><br>RAZ/WI | xxxx       |
| [1]  | E[1]       | Architecturally executed.<br><br>When the PE supports sampling of speculative instructions:<br><b>0b0</b><br>Architecturally executed event is ignored.<br><b>0b1</b><br>Do not record samples that have the Architecturally executed event == 0.                                | x          |
| [0]  | RAZ/<br>WI | Reserved                                                                                                                                                                                                                                                                         | RAZ/<br>WI |

## Access

MRS <Xt>, PMSEVFR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b101 |

MSR PMSEVFR\_EL1, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b101 |

## Accessibility

MRS <Xt>, PMSEVFR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.PMSEVFR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPMS == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMSEVFR_EL1;
    elsif PSTATE.EL == EL2 then
        if MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMSEVFR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = PMSEVFR_EL1;

```

MSR PMSEVFR\_EL1, <Xt>

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGWTR_EL2.PMSEVFR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPMS == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            PMSEVFR_EL1 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                PMSEVFR_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        PMSEVFR_EL1 = X[t, 64];

```

A.15.3 PMSIDR\_EL1, Sampling Profiling ID Register

Describes the Statistical Profiling implementation to software

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Statistical Profiling Extension registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0011 | 0110 | 0100 | xx01 | 0xxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-147: AArch64\_pmsidr\_el1 bit assignments

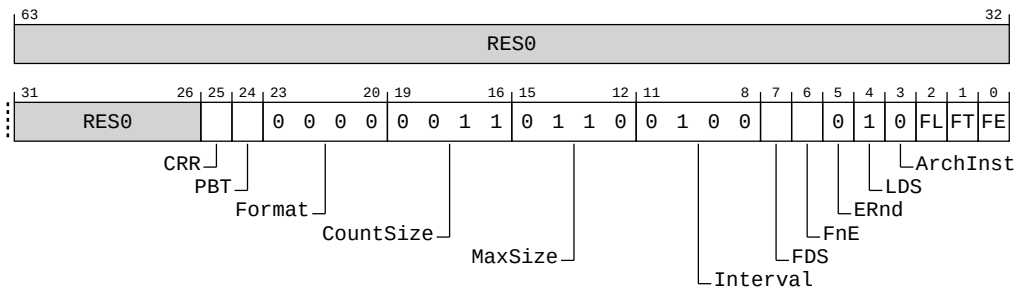


Table A-405: PMSIDR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:26] | RES0 | Reserved    | RES0  |

| Bits    | Name      | Description                                                                                                                                                                                                                                                                                                                   | Reset                                                                     |
|---------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| [25]    | CRR       | Call Return branch records. Defined values are:<br><br><b>0b0</b><br>Operation Type packets for branches do not contain Call Return information.<br><br><b>0b1</b><br>Operation Type packets for branches contain Call Return information.                                                                                    | The reset values can be the following: 0b0, 0b1, respective to the value. |
| [24]    | PBT       | Previous branch target Address packet. Defined values are:<br><br><b>0b0</b><br>Previous branch target Address packet not supported.<br><br><b>0b1</b><br>Previous branch target Address packet support implemented.                                                                                                          | The reset values can be the following: 0b0, 0b1, respective to the value. |
| [23:20] | Format    | Defines the format of the sample records. Defined values are:<br><br><b>0b0000</b><br>Format 0.                                                                                                                                                                                                                               | 0b0000                                                                    |
| [19:16] | CountSize | Defines the size of the counters.<br><br><b>0b0011</b><br>16-bit saturating counters.                                                                                                                                                                                                                                         | 0b0011                                                                    |
| [15:12] | MaxSize   | Defines the largest size for a single record, rounded up to a power-of-two. If this is the same as the minimum alignment (AArch64-PMBIDR_EL1.Align), then each record is exactly this size. Defined values are:<br><br><b>0b0110</b><br>64 bytes.                                                                             | 0b0110                                                                    |
| [11:8]  | Interval  | Recommended minimum sampling interval. This provides guidance from the implementer to the smallest minimum sampling interval, N. Defined values are:<br><br><b>0b0100</b><br>1,024.                                                                                                                                           | 0b0100                                                                    |
| [7]     | FDS       | Filter by data source. Defined values are:<br><br><b>0b0</b><br>AArch64-PMSDSFR_EL1 is not implemented and AArch64-PMSFCR_EL1.FDS is <b>RES0</b> .<br><br><b>0b1</b><br>AArch64-PMSDSFR_EL1 and AArch64-PMSFCR_EL1.FDS are implemented.                                                                                       | The reset values can be the following: 0b0, 0b1, respective to the value. |
| [6]     | FnE       | Filtering by events, inverted. Defined values are:<br><br><b>0b0</b><br>AArch64-PMSNEVFR_EL1 is not implemented and AArch64-PMSFCR_EL1.FnE is <b>RES0</b> .<br><br><b>0b1</b><br>AArch64-PMSNEVFR_EL1 and AArch64-PMSFCR_EL1.FnE are implemented.                                                                             | The reset values can be the following: 0b0, 0b1, respective to the value. |
| [5]     | ERnd      | Defines how the random number generator is used in determining the interval between samples, when enabled by AArch64-PMSIRR_EL1.RND. Defined values are:<br><br><b>0b0</b><br>The random number is added at the start of the interval, and the sample is taken and a new interval started when the combined interval expires. | 0b0                                                                       |

| Bits | Name     | Description                                                                                                                   | Reset |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| [4]  | LDS      | Data source indicator for sampled load instructions. Defined values are:<br><br><b>0b1</b><br>Loaded data source implemented. | 0b1   |
| [3]  | ArchInst | Architectural instruction profiling. Defined values are:<br><br><b>0b0</b><br>Micro-op sampling implemented.                  | 0b0   |
| [2]  | FL       | Filtering by latency. This bit reads as one.                                                                                  | x     |
| [1]  | FT       | Filtering by operation type. This bit reads as one.                                                                           | x     |
| [0]  | FE       | Filtering by events. This bit reads as one.                                                                                   | x     |

## Access

MRS <Xt>, PMSIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b111 |

## Accessibility

MRS <Xt>, PMSIDR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.PMSIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPMS == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = PMSIDR_EL1;
    elsif PSTATE.EL == EL2 then
        if MDCR_EL3.NSPB[0] == '0' || MDCR_EL3.NSPB[1] != SCR_EL3.NS then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = PMSIDR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = PMSIDR_EL1;

```



A.15.4 PMBIDR\_EL1, Profiling Buffer ID Register

Provides information to software as to whether the buffer can be programmed at the current Exception level.

Configurations

This register is available in all configurations.

Attributes

Width

64

Functional group

Statistical Profiling Extension registers

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xx10 | 0110 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure A-148: AArch64\_pmbidr\_el1 bit assignments

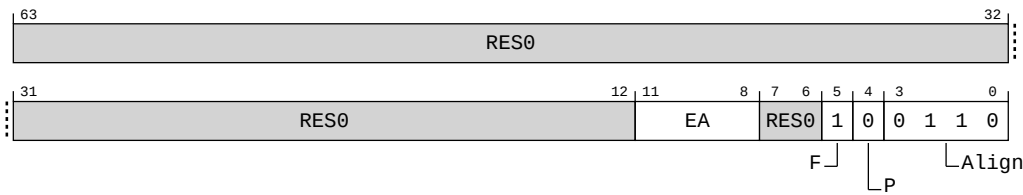


Table A-407: PMBIDR\_EL1 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:12] | RES0 | Reserved    | RES0  |

| Bits   | Name  | Description                                                                                                                                                                                                                                                                                                                                                                         | Reset                                                                                   |
|--------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| [11:8] | EA    | External Abort handling. Describes how the PE manages External aborts on writes made by the Statistical Profiling Unit to the Profiling Buffer.<br><br><b>0b0000</b><br>Not described.<br><br><b>0b0001</b><br>The PE ignores External aborts on writes made by the Statistical Profiling Unit.<br><br><b>0b0010</b><br>The External abort generates an SError interrupt at the PE. | The reset values can be the following: 0b0000, 0b0001, 0b0010, respective to the value. |
| [7:6]  | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                            | RES0                                                                                    |
| [5]    | F     | Flag updates. Describes how address translations performed by the Statistical Profiling Unit manage the Access flag and dirty state.<br><br><b>0b1</b><br>Hardware management of the Access flag and dirty state for accesses made by the Statistical Profiling Unit is controlled in the same way as explicit memory accesses in the Profiling Buffer owning translation regime.   | 0b1                                                                                     |
| [4]    | P     | Programming not allowed. When read at EL3, this field reads as zero. Otherwise, indicates that the Profiling Buffer is owned by a higher Exception level or another Security state. Defined values are:<br><br><b>0b0</b><br>Programming is allowed.                                                                                                                                | 0b0                                                                                     |
| [3:0]  | Align | Defines the minimum alignment constraint for writes to AArch64-PMBPTR_EL1. Defined values are:<br><br><b>0b0110</b><br>64 bytes.                                                                                                                                                                                                                                                    | 0b0110                                                                                  |

## Access

MRS <Xt>, PMBIDR\_EL1

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b11 | 0b000 | 0b1001 | 0b1010 | 0b111 |

## Accessibility

MRS <Xt>, PMBIDR\_EL1

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && SCR_EL3.FGTEn == '1' && HDFGRTR_EL2.PMBIDR_EL1 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = PMBIDR_EL1;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = PMBIDR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = PMBIDR_EL1;

```

## A.16 AArch64 Trace Buffer Extension registers summary

The following summary table provides an overview of all Trace Buffer Extension registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table A-409: Trace Buffer Extension registers summary**

| Name          | Op0 | Op1 | CRn | CRm | Op2 | Reset                      | Width  | Description                            |
|---------------|-----|-----|-----|-----|-----|----------------------------|--------|----------------------------------------|
| TRBLIMITR_EL1 | 3   | 0   | C9  | C11 | 0   | See individual bit resets. | 64-bit | Trace Buffer Limit Address Register    |
| TRBPTR_EL1    | 3   | 0   | C9  | C11 | 1   | See individual bit resets. | 64-bit | Trace Buffer Write Pointer Register    |
| TRBBASER_EL1  | 3   | 0   | C9  | C11 | 2   | See individual bit resets. | 64-bit | Trace Buffer Base Address Register     |
| TRBSR_EL1     | 3   | 0   | C9  | C11 | 3   | See individual bit resets. | 64-bit | Trace Buffer Status/syndrome Register  |
| TRBMAR_EL1    | 3   | 0   | C9  | C11 | 4   | See individual bit resets. | 64-bit | Trace Buffer Memory Attribute Register |
| TRBTRG_EL1    | 3   | 0   | C9  | C11 | 6   | See individual bit resets. | 64-bit | Trace Buffer Trigger Counter Register  |
| TRBIDR_EL1    | 3   | 0   | C9  | C11 | 7   | See individual bit resets. | 64-bit | Trace Buffer ID Register               |

# Appendix B External registers

This appendix contains the descriptions for the Cortex®-X925 external registers.

This manual does not provide a complete list of registers. Read this manual together with the [Arm® Architecture Reference Manual for A-profile architecture](#).

## B.1 External CoreROM registers summary

The following summary table provides an overview of all memory-mapped CoreROM registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table B-1: CoreROM registers summary**

| Offset | Name                               | Reset                      | Width  | Description                                         |
|--------|------------------------------------|----------------------------|--------|-----------------------------------------------------|
| 0x000  | <a href="#">COREROM_ROMENTRY0</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 0                              |
| 0x004  | <a href="#">COREROM_ROMENTRY1</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 1                              |
| 0x008  | <a href="#">COREROM_ROMENTRY2</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 2                              |
| 0x00C  | <a href="#">COREROM_ROMENTRY3</a>  | See individual bit resets. | 32-bit | Core ROM table Entry 3                              |
| 0xFB8  | <a href="#">COREROM_AUTHSTATUS</a> | See individual bit resets. | 32-bit | Core ROM table Authentication Status Register       |
| 0xFBC  | <a href="#">COREROM_DEVARCH</a>    | See individual bit resets. | 32-bit | Core ROM table Device Architecture Register         |
| 0xFCC  | <a href="#">COREROM_DEVTYPE</a>    | See individual bit resets. | 32-bit | Core ROM table Device Type Register                 |
| 0xFD0  | <a href="#">COREROM_PIDR4</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 4 |
| 0xFE0  | <a href="#">COREROM_PIDR0</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 0 |
| 0xFE4  | <a href="#">COREROM_PIDR1</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 1 |
| 0xFE8  | <a href="#">COREROM_PIDR2</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 2 |
| 0xFEC  | <a href="#">COREROM_PIDR3</a>      | See individual bit resets. | 32-bit | Core ROM table Peripheral Identification Register 3 |
| 0xFF0  | <a href="#">COREROM_CIDR0</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 0  |
| 0xFF4  | <a href="#">COREROM_CIDR1</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 1  |
| 0xFF8  | <a href="#">COREROM_CIDR2</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 2  |
| 0xFFC  | <a href="#">COREROM_CIDR3</a>      | See individual bit resets. | 32-bit | Core ROM table Component Identification Register 3  |

B.1.1 COREROM\_ROMENTRY0, Core ROM table Entry 0

Provides the address offset for one CoreSight component.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0x000

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0000 | 0000 | 0000 | 0001 | 0000 | xxxx | xxxx | x011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-1: ext\_corerom\_romentry0 bit assignments

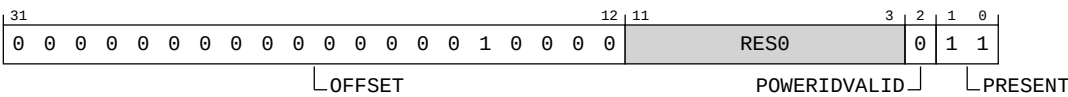


Table B-2: COREROM\_ROMENTRY0 bit descriptions

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                              | Reset   |
|---------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [31:12] | OFFSET | The component address, relative to the base address of this ROM Table. The component address is calculated using the following equation:<br><br>Component Address = ROM Table Base Address + (OFFSET << 12).<br><br><b>0b00000000000000000000000000000000</b><br>Core DBG component at address 0x1_0000. | 0x00010 |
| [11:3]  | RES0   | Reserved                                                                                                                                                                                                                                                                                                 | RES0    |

| Bits  | Name         | Description                                                                                                                | Reset |
|-------|--------------|----------------------------------------------------------------------------------------------------------------------------|-------|
| [2]   | POWERIDVALID | Indicates if the Power domain ID field contains a Power domain ID.<br><br><b>0b0</b><br>A power domain ID is not provided. | 0b0   |
| [1:0] | PRESENT      | Indicates whether an entry is present at this location in the ROM Table.<br><br><b>0b11</b><br>The ROM Entry is present.   | 0b11  |

### Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0x000  | None  |

This interface is accessible as follows:

RO

## B.1.2 COREROM\_ROMENTRY1, Core ROM table Entry 1

Provides the address offset for one CoreSight component.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Component

CoreROM

#### Register offset

0x004

#### Access type

RO

#### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0000 | 0000 | 0000 | 0010 | 0000 | xxxx | xxxx | x011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.



Register offset

0x008

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0000 | 0000 | 0000 | 0011 | 0000 | xxxx | xxxx | x011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-3: ext\_corerom\_romentry2 bit assignments

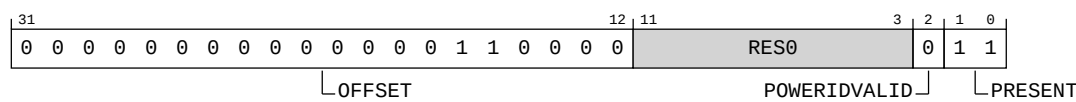


Table B-6: COREROM\_ROMENTRY2 bit descriptions

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                   | Reset   |
|---------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [31:12] | OFFSET       | The component address, relative to the base address of this ROM Table. The component address is calculated using the following equation:<br><br>Component Address = ROM Table Base Address + (OFFSET << 12).<br><br><b>0b000000000000000110000</b><br>Core ETM component at address 0x3_0000. | 0x00030 |
| [11:3]  | RES0         | Reserved                                                                                                                                                                                                                                                                                      | RES0    |
| [2]     | POWERIDVALID | Indicates if the Power domain ID field contains a Power domain ID.<br><br><b>0b0</b><br>A power domain ID is not provided.                                                                                                                                                                    | 0b0     |
| [1:0]   | PRESENT      | Indicates whether an entry is present at this location in the ROM Table.<br><br><b>0b11</b><br>The ROM Entry is present.                                                                                                                                                                      | 0b11    |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0x008  | None  |

This interface is accessible as follows:



RO

B.1.4 COREROM\_ROMENTRY3, Core ROM table Entry 3

Provides the address offset for one CoreSight component.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0x00C

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0000 | 0000 | 0000 | 0100 | 0000 | xxxx | xxxx | x011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-4: ext\_corerom\_romentry3 bit assignments

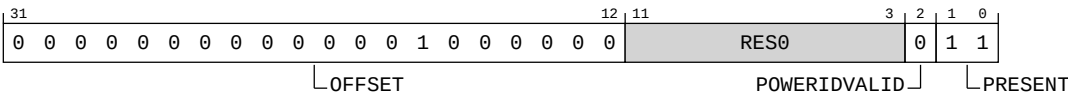


**Table B-8: COREROM\_ROMENTRY3 bit descriptions**

| Bits    | Name         | Description                                                                                                                                                                                                                                                                                                                                                            | Reset   |
|---------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [31:12] | OFFSET       | The component address, relative to the base address of this ROM Table. The component address is calculated using the following equation:<br><br>Component Address = ROM Table Base Address + (OFFSET << 12).<br><br><b>0b000000000000001000000</b><br><br>Core ELA component at address 0x4_0000. When the core is configured without ELA, this field is set to 0x000. | 0x00040 |
| [11:3]  | RES0         | Reserved                                                                                                                                                                                                                                                                                                                                                               | RES0    |
| [2]     | POWERIDVALID | Indicates if the Power domain ID field contains a Power domain ID.<br><br><b>0b0</b><br><br>A power domain ID is not provided.                                                                                                                                                                                                                                         | 0b0     |
| [1:0]   | PRESENT      | Indicates whether an entry is present at this location in the ROM Table.<br><br><b>0b11</b><br><br>The ROM Entry is present. When the core is configured without ELA, this field is set to 0x0.                                                                                                                                                                        | 0b11    |

### Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0x00C  | None  |

This interface is accessible as follows:

RO

## B.1.5 COREROM\_AUTHSTATUS, Core ROM table Authentication Status Register

Provides information about the state of the authentication interface for debug.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Component

CoreROM

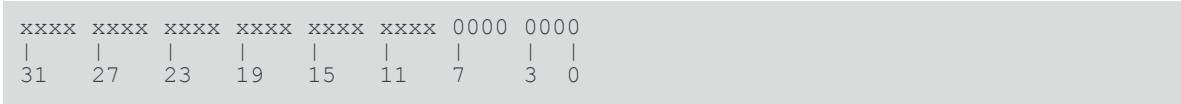
#### Register offset

0xFB8

#### Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-5: ext\_corerom\_authstatus bit assignments

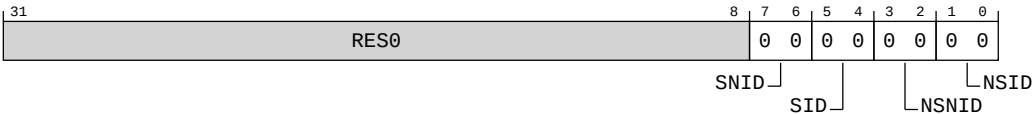


Table B-10: COREROM\_AUTHSTATUS bit descriptions

| Bits   | Name  | Description                                                                     | Reset |
|--------|-------|---------------------------------------------------------------------------------|-------|
| [31:8] | RES0  | Reserved                                                                        | RES0  |
| [7:6]  | SNID  | Secure Non-invasive Debug.<br><br>0b00<br><br>This field is not supported       | 0b00  |
| [5:4]  | SID   | Secure Invasive Debug.<br><br>0b00<br><br>This field is not supported           | 0b00  |
| [3:2]  | NSNID | Non-secure Non-invasive Debug.<br><br>0b00<br><br>Debug level is not supported. | 0b00  |
| [1:0]  | NSID  | Non-secure Invasive Debug.<br><br>0b00<br><br>Debug level is not supported.     | 0b00  |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFB8  | None  |

This interface is accessible as follows:

RO

B.1.6 COREROM\_DEVARCH, Core ROM table Device Architecture Register

Identifies the architect and architecture of a CoreSight component.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0xFBC

Access type

RO

Reset value

0100 0111 0111 0000 0000 1010 1111 0111

Bit descriptions

Figure B-6: ext\_corerom\_devarch bit assignments

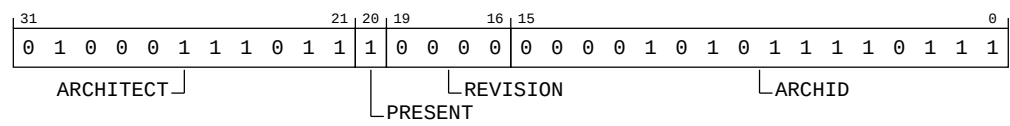


Table B-12: COREROM\_DEVARCH bit descriptions

| Bits    | Name      | Description                                                                                    | Reset         |
|---------|-----------|------------------------------------------------------------------------------------------------|---------------|
| [31:21] | ARCHITECT | Architect.<br><b>0b01000111011</b><br>JEP106 continuation code 0x4, ID code 0x3B. Arm Limited. | 0b01000111011 |
| [20]    | PRESENT   | Present.<br><b>0b1</b><br>DEVARCH information present.                                         | 0b1           |
| [19:16] | REVISION  | Revision.<br><b>0b0000</b><br>Revision 0.                                                      | 0b0000        |

| Bits   | Name   | Description                                                                                                                                                                                        | Reset  |
|--------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:0] | ARCHID | Architecture ID.<br><br><b>0b0000101011110111</b><br><br>ROM Table v0. The debug tool must inspect ext-COREROM_DEVTYPE and ext-COREROM_DEVID to determine further information about the ROM Table. | 0x0AF7 |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFBC  | None  |

This interface is accessible as follows:

RO

B.1.7 COREROM\_DEVTYPE, Core ROM table Device Type Register

A debugger can use DEVTYPE to obtain information about a component that has an unrecognized part number.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0xFCC

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-7: ext\_corerom\_devtype bit assignments



Table B-14: COREROM\_DEVTYPE bit descriptions

| Bits   | Name  | Description                                      | Reset  |
|--------|-------|--------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                         | RES0   |
| [7:4]  | SUB   | Sub number<br><b>0b0000</b><br>Other, undefined. | 0b0000 |
| [3:0]  | MAJOR | Major number<br><b>0b0000</b><br>Miscellaneous.  | 0b0000 |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFCC  | None  |

This interface is accessible as follows:

RO

B.1.8 COREROM\_PIDR4, Core ROM table Peripheral Identification Register 4

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

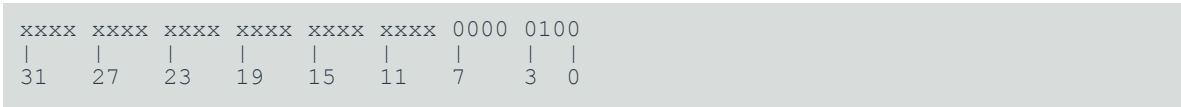
CoreROM

Register offset

0xFD0

Access type  
RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-8: ext\_corerom\_pidr4 bit assignments

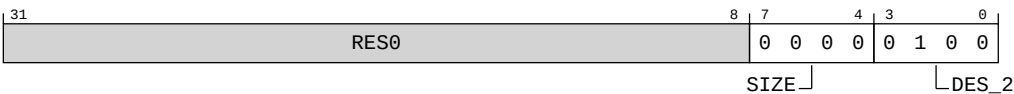


Table B-16: COREROM\_PIDR4 bit descriptions

| Bits   | Name  | Description                                                                                                                   | Reset  |
|--------|-------|-------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                      | RES0   |
| [7:4]  | SIZE  | 4KB count.<br><b>0b0000</b><br>The component uses a single 4KB block.                                                         | 0b0000 |
| [3:0]  | DES_2 | JEP106 continuation code.<br><b>0b0100</b><br>Arm Limited. Number of 0x7F bytes in full JEP106 code 0x7F 0x7F 0x7F 0x7F 0x3B. | 0b0100 |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFD0  | None  |

This interface is accessible as follows:

RO

### B.1.9 COREROM\_PIDR0, Core ROM table Peripheral Identification Register 0

Provides CoreSight discovery information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

CoreROM

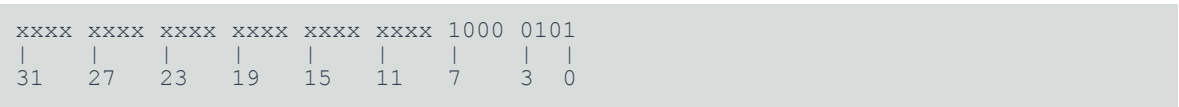
##### Register offset

0xFE0

##### Access type

RO

##### Reset value



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-9: ext\_corerom\_pidr0 bit assignments

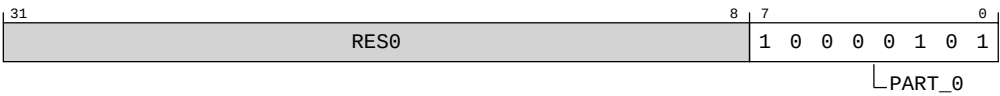


Table B-18: COREROM\_PIDR0 bit descriptions

| Bits   | Name   | Description                                                                                                  | Reset |
|--------|--------|--------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0   | Reserved                                                                                                     | RES0  |
| [7:0]  | PART_0 | Part number bits [7:0].<br><b>0b10000101</b><br>Cortex-X925 Core ROM table. Bits [7:0] of part number 0xD85. | 0x85  |



Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFE0  | None  |

This interface is accessible as follows:

RO

B.1.10 COREROM\_PIDR1, Core ROM table Peripheral Identification Register 1

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0xFE4

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1011 | 1101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-10: ext\_corerom\_pidr1 bit assignments

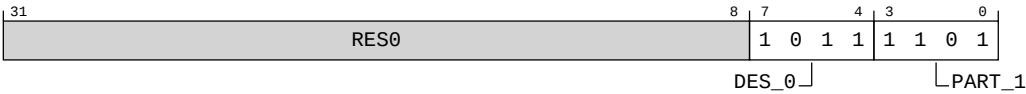


Table B-20: COREROM\_PIDR1 bit descriptions

| Bits   | Name   | Description                                                                                                                | Reset  |
|--------|--------|----------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                                                   | RES0   |
| [7:4]  | DES_0  | JEP106 identification code bits [3:0].<br><br><b>0b1011</b><br>Arm Limited. Bits [3:0] of JEP106 identification code 0x3B. | 0b1011 |
| [3:0]  | PART_1 | Part number bits [11:8].<br><br><b>0b1101</b><br>Cortex-X925 Core ROM table. Bits [11:8] of part number 0xD85.             | 0b1101 |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFE4  | None  |

This interface is accessible as follows:

RO

B.1.11 COREROM\_PIDR2, Core ROM table Peripheral Identification Register 2

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0xFE8

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-11: ext\_corerom\_pidr2 bit assignments

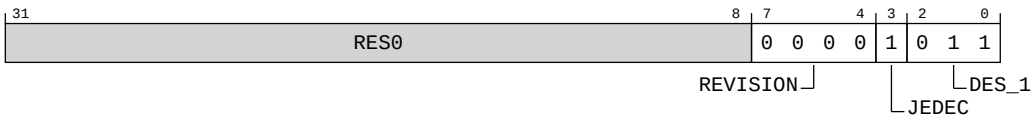


Table B-22: COREROM\_PIDR2 bit descriptions

| Bits   | Name     | Description                                                                                                        | Reset  |
|--------|----------|--------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0     | Reserved                                                                                                           | RES0   |
| [7:4]  | REVISION | Component revision.<br><br>0b0000<br>rOp2                                                                          | 0b0000 |
| [3]    | JEDEC    | JEDEC assignee.<br><br>0b1<br>JEDEC-assignee values is used.                                                       | 0b1    |
| [2:0]  | DES_1    | JEP106 identification code bits [6:4].<br><br>0b011<br>Arm Limited. Bits [6:4] of JEP106 identification code 0x3B. | 0b011  |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFE8  | None  |

This interface is accessible as follows:

RO

B.1.12 COREROM\_PIDR3, Core ROM table Peripheral Identification Register 3

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

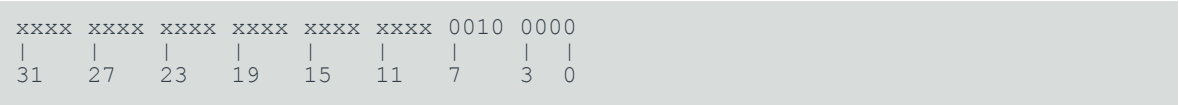
Register offset

0xFEC

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-12: ext\_corerom\_pidr3 bit assignments



Table B-24: COREROM\_PIDR3 bit descriptions

| Bits   | Name   | Description                                                                             | Reset  |
|--------|--------|-----------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                | RES0   |
| [7:4]  | REVAND | Minor errata fixes.<br>0b0010                                                           | 0b0010 |
| [3:0]  | CMOD   | Customer Modified.<br>0b0000<br>The component is not modified from the original design. | 0b0000 |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFEC  | None  |

This interface is accessible as follows:

RO

B.1.13 COREROM\_CIDR0, Core ROM table Component Identification Register 0

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0xFF0

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-13: ext\_corerom\_cidr0 bit assignments

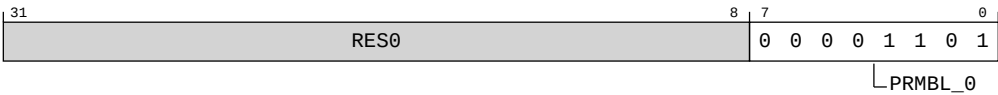


Table B-26: COREROM\_CIDR0 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:8] | RES0 | Reserved    | RES0  |

| Bits  | Name    | Description                                                                                                           | Reset |
|-------|---------|-----------------------------------------------------------------------------------------------------------------------|-------|
| [7:0] | PRMBL_0 | CoreSight component identification preamble.<br><br><b>0b00001101</b><br>CoreSight component identification preamble. | 0x0D  |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFF0  | None  |

This interface is accessible as follows:

RO

B.1.14 COREROM\_CIDR1, Core ROM table Component Identification Register 1

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

CoreROM

Register offset

0xFF4

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1001 | 0000 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-14: ext\_corerom\_cidr1 bit assignments

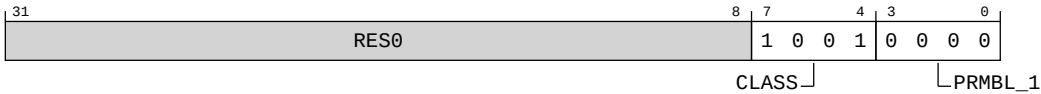


Table B-28: COREROM\_CIDR1 bit descriptions

| Bits   | Name    | Description                                                                                                   | Reset  |
|--------|---------|---------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0    | Reserved                                                                                                      | RES0   |
| [7:4]  | CLASS   | CoreSight component class.<br><b>0b1001</b><br>CoreSight component.                                           | 0b1001 |
| [3:0]  | PRMBL_1 | CoreSight component identification preamble.<br><b>0b0000</b><br>CoreSight component identification preamble. | 0b0000 |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFF4  | None  |

This interface is accessible as follows:

RO

B.1.15 COREROM\_CIDR2, Core ROM table Component Identification Register 2

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

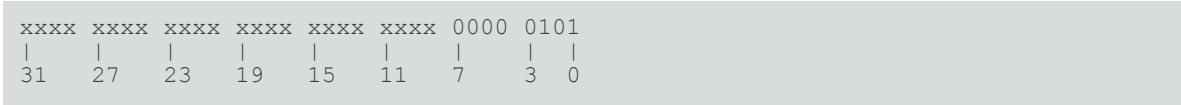
CoreROM

Register offset

0xFF8

Access type  
RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-15: ext\_corerom\_cidr2 bit assignments



Table B-30: COREROM\_CIDR2 bit descriptions

| Bits   | Name    | Description                                                                                                        | Reset |
|--------|---------|--------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0    | Reserved                                                                                                           | RES0  |
| [7:0]  | PRMBL_2 | CoreSight component identification preamble.<br><b>0b000000101</b><br>CoreSight component identification preamble. | 0x05  |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFF8  | None  |

This interface is accessible as follows:

RO

B.1.16 COREROM\_CIDR3, Core ROM table Component Identification Register 3

Provides CoreSight discovery information.

Configurations

This register is available in all configurations.



Attributes

Width

32

Component

CoreROM

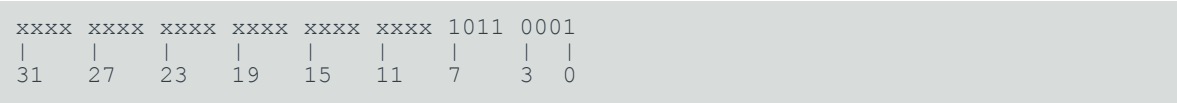
Register offset

0xFFC

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-16: ext\_corerom\_cidr3 bit assignments



Table B-32: COREROM\_CIDR3 bit descriptions

| Bits   | Name    | Description                                                                                                       | Reset |
|--------|---------|-------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0    | Reserved                                                                                                          | RES0  |
| [7:0]  | PRMBL_3 | CoreSight component identification preamble.<br><b>0b10110001</b><br>CoreSight component identification preamble. | 0xB1  |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| CoreROM   | 0xFFC  | None  |

This interface is accessible as follows:

RO

## B.2 External PPM registers summary

The following summary table provides an overview of all memory-mapped PPM registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

Table B-34: PPM registers summary

| Offset | Name                       | Reset                      | Width  | Description                                                    |
|--------|----------------------------|----------------------------|--------|----------------------------------------------------------------|
| 0x000  | <a href="#">CPUPPMCR</a>   | See individual bit resets. | 64-bit | Global Performance and Power Management Configuration Register |
| 0x010  | <a href="#">CPUMPMCR</a>   | See individual bit resets. | 64-bit | Global MPMM Control Register                                   |
| 0x020  | <a href="#">CPUPMPDPCR</a> | See individual bit resets. | 64-bit | Performance and Power Management PDP Control Register          |
| 0x080  | <a href="#">CPUPPMCR4</a>  | See individual bit resets. | 64-bit | Power Performance Management Register                          |
| 0x088  | <a href="#">CPUPPMCR5</a>  | See individual bit resets. | 64-bit | Power Performance Management Register                          |
| 0x090  | <a href="#">CPUPPMCR6</a>  | See individual bit resets. | 64-bit | Power Performance Management Register                          |

### B.2.1 CPUPPMCR, Global Performance and Power Management Configuration Register

Provides **IMPLEMENTATION DEFINED** control and discovery of the Performance and Power Management (PPM) features.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Component

PPM

##### Register offset

0x000

##### Access type

RW

##### Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | x011 | xxxx | x011 | xxxx | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |

**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure B-17: ext\_cpuppmcr bit assignments**

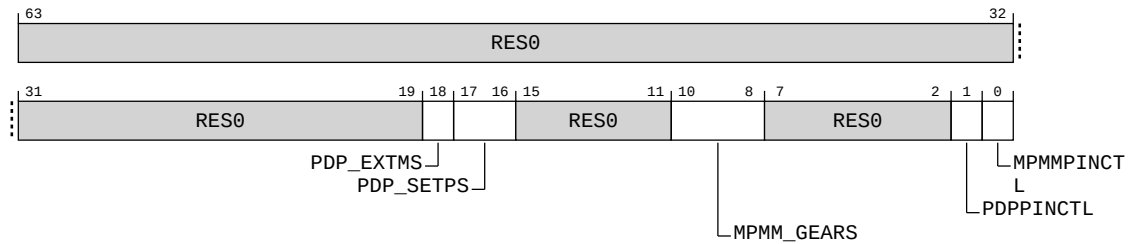


**Table B-35: CPUPPMCR bit descriptions**

| Bits    | Name       | Description                                                                                                                       | Reset |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:19] | RES0       | Reserved                                                                                                                          | RES0  |
| [18]    | PDP_EXTMS  | External memory system PDP control<br><b>0b0</b><br>Independent external memory system PDP control is not implemented             | 0b0   |
| [17:16] | PDP_SETPS  | Number of PDP Setpoints Implemented<br><b>0b11</b><br>3 PDP setpoints are implemented                                             | 0b11  |
| [15:11] | RES0       | Reserved                                                                                                                          | RES0  |
| [10:8]  | MPMM_GEARs | Number of MPMM Gears Implemented<br><b>0b011</b><br>3 MPMM gears are implemented                                                  | 0b011 |
| [7:2]   | RES0       | Reserved                                                                                                                          | RES0  |
| [1]     | PDPPINCTL  | PDP Pin Control Enabled<br><b>0b0</b><br>PDP control through SPR and utility bus<br><b>0b1</b><br>PDP control through pin only    | 0b0   |
| [0]     | MPMPINCTL  | MPMM Pin Control Enabled<br><b>0b0</b><br>MPMM control through SPR and utility bus<br><b>0b1</b><br>MPMM control through pin only | 0b0   |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PPM       | 0x000  | None  |

This interface is accessible as follows:

RW

B.2.2 CPUMPMCR, Global MPMM Control Register

Provides **IMPLEMENTATION DEFINED** control of the Maximum Power Mitigation Mechanism (MPMM) feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

PPM

Register offset

0x010

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | x000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-18: ext\_cpumpmmcr bit assignments

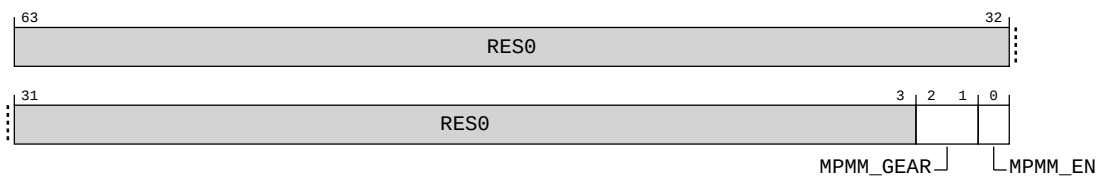


Table B-37: CPUMPMMCR bit descriptions

| Bits   | Name      | Description                                                                                                                                                                          | Reset |
|--------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:3] | RES0      | Reserved                                                                                                                                                                             | RES0  |
| [2:1]  | MPMM_GEAR | MPMM Gear Select<br><br><b>0b00</b><br>Select MPMM Gear 0<br><br><b>0b01</b><br>Select MPMM Gear 1<br><br><b>0b10</b><br>Select MPMM Gear 2<br><br><b>0b11</b><br>Select MPMM Gear 3 | 0b00  |
| [0]    | MPMM_EN   | MPMM Global Enable<br><br><b>0b0</b><br>MPMM is disabled<br><br><b>0b1</b><br>MPMM is enabled                                                                                        | 0b0   |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PPM       | 0x010  | None  |

This interface is accessible as follows:

RW

B.2.3 CPUPMPDPCR, Performance and Power Management PDP Control Register

Provides **IMPLEMENTATION DEFINED** control of the Performance Defined Power (PDP) feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

PPM

Register offset

0x020

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-19: ext\_cpupmpdpcr bit assignments

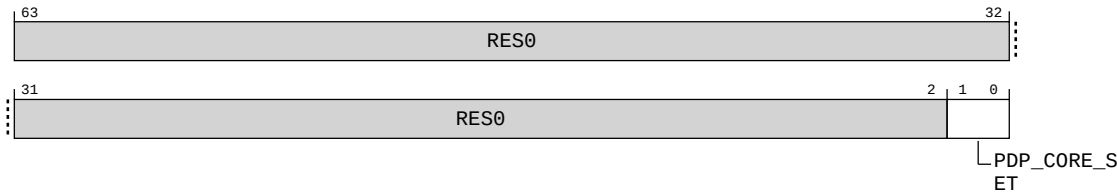


Table B-39: CPUPMPDPCR bit descriptions

| Bits   | Name         | Description                                                                                                                                                                                                                        | Reset |
|--------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:2] | RES0         | Reserved                                                                                                                                                                                                                           | RES0  |
| [1:0]  | PDP_CORE_SET | Core PDP Aggressiveness<br><br><b>0b00</b><br>Disable PDP<br><br><b>0b01</b><br>Enable PDP at low aggressiveness<br><br><b>0b10</b><br>Enable PDP at medium aggressiveness<br><br><b>0b11</b><br>Enable PDP at high aggressiveness | 0b00  |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PPM       | 0x020  | None  |

This interface is accessible as follows:

RW

B.2.4 CPUPPMCR4, Power Performance Management Register

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

PPM

Register offset

0x080

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-20: ext\_cpuppmcr4 bit assignments

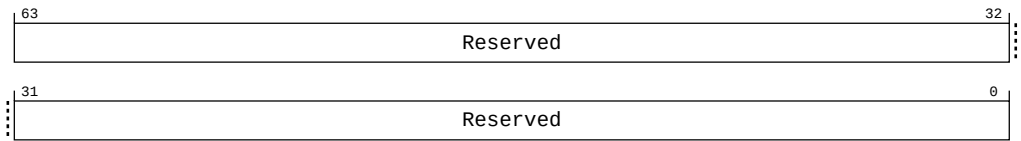


Table B-41: CPUPPMCR4 bit descriptions

| Bits   | Name     | Description                    | Reset  |
|--------|----------|--------------------------------|--------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 {x} |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PPM       | 0x080  | None  |

This interface is accessible as follows:

RW

B.2.5 CPUPPMCR5, Power Performance Management Register

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

PPM

Register offset

0x088

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Reserved



Figure B-21: ext\_cpuppmcr5 bit assignments

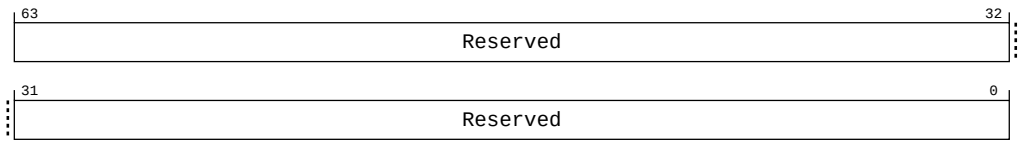


Table B-43: CPUPPMCR5 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PPM       | 0x088  | None  |

This interface is accessible as follows:

RW

B.2.6 CPUPPMCR6, Power Performance Management Register

This register contains control bits that affect the CPU behavior

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

PPM

Register offset

0x090

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Reserved

Figure B-22: ext\_cpuppmcr6 bit assignments

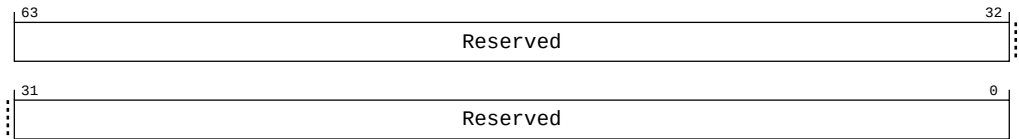


Table B-45: CPUPPMCR6 bit descriptions

| Bits   | Name     | Description                    | Reset    |
|--------|----------|--------------------------------|----------|
| [63:0] | Reserved | Reserved for Arm internal use. | 64 { x } |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PPM       | 0x090  | None  |

This interface is accessible as follows:

RW

B.3 External PMU registers summary

The following summary table provides an overview of all memory-mapped PMU registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table B-47: PMU registers summary**

| Offset | Name                  | Reset                      | Width  | Description                                |
|--------|-----------------------|----------------------------|--------|--------------------------------------------|
| 0x0    | PMEVCNTR0_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x8    | PMEVCNTR1_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x10   | PMEVCNTR2_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x18   | PMEVCNTR3_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x20   | PMEVCNTR4_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x28   | PMEVCNTR5_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x30   | PMEVCNTR6_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x38   | PMEVCNTR7_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x40   | PMEVCNTR8_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x48   | PMEVCNTR9_ELO [63:0]  | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x50   | PMEVCNTR10_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x58   | PMEVCNTR11_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x60   | PMEVCNTR12_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x68   | PMEVCNTR13_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x70   | PMEVCNTR14_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x78   | PMEVCNTR15_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x80   | PMEVCNTR16_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x88   | PMEVCNTR17_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x90   | PMEVCNTR18_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x98   | PMEVCNTR19_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xA0   | PMEVCNTR20_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xA8   | PMEVCNTR21_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xB0   | PMEVCNTR22_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xB8   | PMEVCNTR23_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xC0   | PMEVCNTR24_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xC8   | PMEVCNTR25_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xD0   | PMEVCNTR26_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xD8   | PMEVCNTR27_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xE0   | PMEVCNTR28_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xE8   | PMEVCNTR29_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0xF0   | PMEVCNTR30_ELO [63:0] | See individual bit resets. | 64-bit | Performance Monitors Event Count Registers |
| 0x0F8  | PMCCNTR_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Cycle Counter         |
| 0x0FC  | PMCCNTR_ELO [63:32]   | See individual bit resets. | 32-bit | Performance Monitors Cycle Counter         |
| 0x200  | PMPCSR [31:0]         | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x204  | PMPCSR [63:32]        | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x220  | PMPCSR [31:0]         | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x224  | PMPCSR [63:32]        | See individual bit resets. | 32-bit | Program Counter Sample Register            |
| 0x208  | PMCID1SR              | See individual bit resets. | 32-bit | CONTEXTIDR_EL1 Sample Register             |
| 0x228  | PMCID1SR              | See individual bit resets. | 32-bit | CONTEXTIDR_EL1 Sample Register             |

| Offset | Name                      | Reset                      | Width  | Description                                        |
|--------|---------------------------|----------------------------|--------|----------------------------------------------------|
| 0x20C  | PMVIDSR                   | See individual bit resets. | 32-bit | VMID Sample Register                               |
| 0x22C  | PMCID2SR                  | See individual bit resets. | 32-bit | CONTEXTIDR_EL2 Sample Register                     |
| 0x400  | PMEVTYPER0_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x404  | PMEVTYPER1_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x408  | PMEVTYPER2_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x40C  | PMEVTYPER3_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x410  | PMEVTYPER4_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x414  | PMEVTYPER5_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x418  | PMEVTYPER6_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x41C  | PMEVTYPER7_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x420  | PMEVTYPER8_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x424  | PMEVTYPER9_ELO [31:0]     | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x428  | PMEVTYPER10_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x42C  | PMEVTYPER11_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x430  | PMEVTYPER12_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x434  | PMEVTYPER13_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x438  | PMEVTYPER14_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x43C  | PMEVTYPER15_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x440  | PMEVTYPER16_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x444  | PMEVTYPER17_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x448  | PMEVTYPER18_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x44C  | PMEVTYPER19_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x450  | PMEVTYPER20_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x454  | PMEVTYPER21_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x458  | PMEVTYPER22_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x45C  | PMEVTYPER23_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x460  | PMEVTYPER24_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x464  | PMEVTYPER25_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x468  | PMEVTYPER26_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x46C  | PMEVTYPER27_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x470  | PMEVTYPER28_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x474  | PMEVTYPER29_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x478  | PMEVTYPER30_ELO [31:0]    | See individual bit resets. | 32-bit | Performance Monitors Event Type Registers          |
| 0x47C  | PMCCFILTR_ELO [31:0]      | See individual bit resets. | 32-bit | Performance Monitors Cycle Counter Filter Register |
| 0x600  | <a href="#">PMPCSSR</a>   | See individual bit resets. | 64-bit | Snapshot Program Counter Sample Register           |
| 0x608  | <a href="#">PMCIDSSR</a>  | See individual bit resets. | 32-bit | Snapshot CONTEXTIDR_EL1 Sample Register            |
| 0x60C  | <a href="#">PMCID2SSR</a> | See individual bit resets. | 32-bit | Snapshot CONTEXTIDR_EL2 Sample Register            |
| 0x610  | <a href="#">PMSSSR</a>    | See individual bit resets. | 32-bit | PMU Snapshot Status Register                       |
| 0x618  | <a href="#">PMCCNTSR</a>  | See individual bit resets. | 64-bit | PMU Cycle Counter Snapshot Register                |
| 0x620  | PMEVCNTRSRO               | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                |

| Offset | Name                          | Reset                      | Width  | Description                                                 |
|--------|-------------------------------|----------------------------|--------|-------------------------------------------------------------|
| 0x628  | PMEVCNTR1                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x630  | PMEVCNTR2                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x638  | PMEVCNTR3                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x640  | PMEVCNTR4                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x648  | PMEVCNTR5                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x650  | PMEVCNTR6                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x658  | PMEVCNTR7                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x660  | PMEVCNTR8                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x668  | PMEVCNTR9                     | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x670  | PMEVCNTR10                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x678  | PMEVCNTR11                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x680  | PMEVCNTR12                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x688  | PMEVCNTR13                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x690  | PMEVCNTR14                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x698  | PMEVCNTR15                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6A0  | PMEVCNTR16                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6A8  | PMEVCNTR17                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6B0  | PMEVCNTR18                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6B8  | PMEVCNTR19                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6C0  | PMEVCNTR20                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6C8  | PMEVCNTR21                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6D0  | PMEVCNTR22                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6D8  | PMEVCNTR23                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6E0  | PMEVCNTR24                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6E8  | PMEVCNTR25                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6F0  | PMEVCNTR26                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x6F8  | PMEVCNTR27                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x700  | PMEVCNTR28                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x708  | PMEVCNTR29                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0x710  | PMEVCNTR30                    | See individual bit resets. | 64-bit | PMU Event Counter Snapshot Register                         |
| 0xC00  | PMCNTENSET_ELO [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Count Enable Set Register              |
| 0xC20  | PMCNTENCLR_ELO [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Count Enable Clear Register            |
| 0xC40  | PMINTENSET_EL1 [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Interrupt Enable Set Register          |
| 0xC60  | PMINTENCLR_EL1 [31:0]         | See individual bit resets. | 32-bit | Performance Monitors Interrupt Enable Clear Register        |
| 0xC80  | PMOVSLR_ELO [31:0]            | See individual bit resets. | 32-bit | Performance Monitors Overflow Flag Status Clear register    |
| 0xCA0  | PMSWINC_ELO                   | See individual bit resets. | 32-bit | Performance Monitors Software Increment Register            |
| 0xCC0  | PMOVSET_ELO [31:0]            | See individual bit resets. | 32-bit | Performance Monitors Overflow Flag Status Set Register      |
| 0xE00  | <a href="#">PMCFGR [31:0]</a> | See individual bit resets. | 32-bit | Performance Monitors Configuration Register                 |
| 0xE04  | <a href="#">PMCR_ELO</a>      | See individual bit resets. | 32-bit | Performance Monitors Control Register                       |
| 0xE20  | <a href="#">PMCEID0</a>       | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 0 |

| Offset | Name                         | Reset                      | Width  | Description                                                 |
|--------|------------------------------|----------------------------|--------|-------------------------------------------------------------|
| 0xE24  | <a href="#">PMCEID1</a>      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 1 |
| 0xE28  | <a href="#">PMCEID2</a>      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 2 |
| 0xE2C  | <a href="#">PMCEID3</a>      | See individual bit resets. | 32-bit | Performance Monitors Common Event Identification register 3 |
| 0xE30  | <a href="#">PMSSCR</a>       | See individual bit resets. | 32-bit | PMU Snapshot Capture Register                               |
| 0xE40  | <a href="#">PMMIR [31:0]</a> | See individual bit resets. | 32-bit | Performance Monitors Machine Identification Register        |
| 0xFA8  | PMDEVAFF0                    | See individual bit resets. | 32-bit | Performance Monitors Device Affinity register 0             |
| 0xFAC  | PMDEVAFF1                    | See individual bit resets. | 32-bit | Performance Monitors Device Affinity register 1             |
| 0xFB0  | PMLAR                        | See individual bit resets. | 32-bit | Performance Monitors Lock Access Register                   |
| 0xFB4  | PMLSR                        | See individual bit resets. | 32-bit | Performance Monitors Lock Status Register                   |
| 0xFB8  | PMAUTHSTATUS                 | See individual bit resets. | 32-bit | Performance Monitors Authentication Status register         |
| 0xFBC  | <a href="#">PMDEVARCH</a>    | See individual bit resets. | 32-bit | Performance Monitors Device Architecture register           |
| 0xFC8  | <a href="#">PMDEVID</a>      | See individual bit resets. | 32-bit | Performance Monitors Device ID register                     |
| 0xFCC  | <a href="#">PMDEVTYPE</a>    | See individual bit resets. | 32-bit | Performance Monitors Device Type register                   |
| 0xFD0  | <a href="#">PMPIDR4</a>      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 4   |
| 0xFE0  | <a href="#">PMPIDR0</a>      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 0   |
| 0xFE4  | <a href="#">PMPIDR1</a>      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 1   |
| 0xFE8  | <a href="#">PMPIDR2</a>      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 2   |
| 0xFEC  | <a href="#">PMPIDR3</a>      | See individual bit resets. | 32-bit | Performance Monitors Peripheral Identification Register 3   |
| 0xFF0  | <a href="#">PMCIDR0</a>      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 0    |
| 0xFF4  | <a href="#">PMCIDR1</a>      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 1    |
| 0xFF8  | <a href="#">PMCIDR2</a>      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 2    |
| 0xFFC  | <a href="#">PMCIDR3</a>      | See individual bit resets. | 32-bit | Performance Monitors Component Identification Register 3    |

### B.3.1 PMPCSSR, Snapshot Program Counter Sample Register

Captured copy of the Program Counter.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Component

PMU

##### Register offset

0x600

Access type

Read

R

Write

RESERVED

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-23: ext\_pmpcssr bit assignments

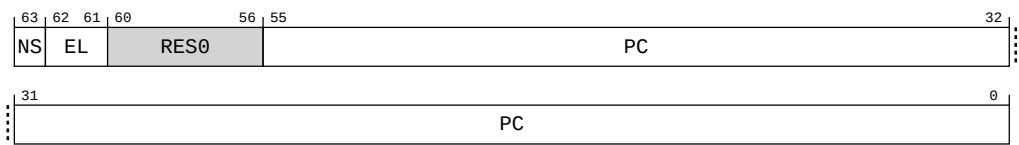


Table B-48: PMPCSSR bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset    |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [63]    | NS   | Non-secure sample.<br><br><b>0b0</b><br>The captured instruction was executed in Secure state.<br><br><b>0b1</b><br>The captured instruction was executed in Non-secure state.                                                                                                                                                                                                                                                                                                                                                     | x        |
| [62:61] | EL   | Exception level sample. The Exception level the captured instruction was executed at.                                                                                                                                                                                                                                                                                                                                                                                                                                              | xx       |
| [60:56] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RES0     |
| [55:0]  | PC   | Sampled PC.<br><br>The instruction address for the sampled instruction. The sampled instruction must be an instruction recently executed by the PE.<br><br>The architecture does not require that all instructions are eligible for sampling.<br><br>The sampled instruction must be architecturally executed. However, in exceptional circumstances, such as a change in security state or other boundary condition, it is permissible to sample an instruction that was speculatively executed and not architecturally executed. | 56 { x } |

Accessibility

PMPCSSR is set to an UNKNOWN value by a read of the EDPCSRlo or PMPCSR[31:0] register.

| Component | Offset | Range |
|-----------|--------|-------|
| PMU       | 0x600  | None  |

This interface is accessible as follows:

RO

B.3.2 PMCIDSSR, Snapshot CONTEXTIDR\_EL1 Sample Register

Captured copy of the CONTEXTIDR\_EL1 register.

The value captured must relate to the instruction captured in PMPCSSR.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

PMU

Register offset

0x608

Access type

Read

R

Write

RESERVED

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.



Bit descriptions

Figure B-24: ext\_pmcidssr bit assignments

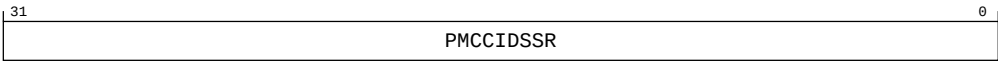


Table B-50: PMCIDSSR bit descriptions

| Bits   | Name      | Description                                      | Reset  |
|--------|-----------|--------------------------------------------------|--------|
| [31:0] | PMCCIDSSR | PMCIDSR sample. Sampled CONTEXTIDR_EL1 snapshot. | 32 {x} |

Accessibility

PMCIDSSR is set to an UNKNOWN value by a read of the EDPCSRlo or PMPCSR[31:0] register.

| Component | Offset | Range |
|-----------|--------|-------|
| PMU       | 0x608  | None  |

This interface is accessible as follows:

RO

B.3.3 PMCID2SSR, Snapshot CONTEXTIDR\_EL2 Sample Register

Captured copy of the CONTEXTIDR\_EL2 register.

The value captured must relate to the instruction captured in PMPCSSR.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

PMU

Register offset

0x60C

Access type

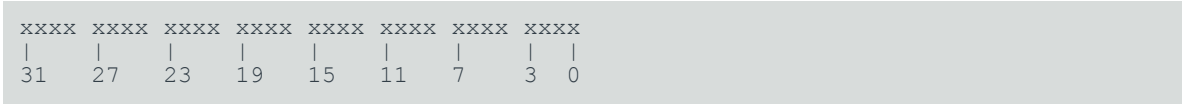
Read

R

Write

RESERVED

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-25: ext\_pmcid2ssr bit assignments

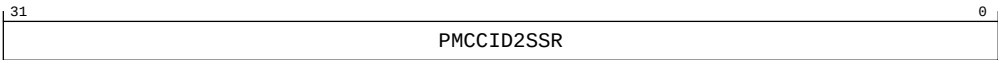


Table B-52: PMCID2SSR bit descriptions

| Bits   | Name       | Description                                       | Reset  |
|--------|------------|---------------------------------------------------|--------|
| [31:0] | PMCCID2SSR | PMCID2SR sample. Sampled CONTEXTIDR_EL2 snapshot. | 32 {x} |

Access

If ARMv8.2 is not implemented, this location is used for PMVIDSSR.

PMCID2SSR is set to an **UNKNOWN** value by a read of the EDPCSRlo or PMPCSR[31:0] register.

Accessibility

If ARMv8.2 is not implemented, this location is used for PMVIDSSR.

PMCID2SSR is set to an UNKNOWN value by a read of the EDPCSRlo or PMPCSR[31:0] register.

| Component | Offset | Range |
|-----------|--------|-------|
| PMU       | 0x60C  | None  |

This interface is accessible as follows:

RO

B.3.4 PMSSSR, PMU Snapshot Status Register

Holds status information about the captured counters.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

PMU

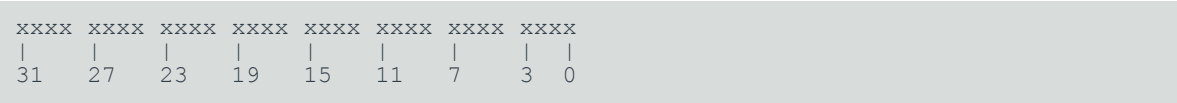
Register offset

0x610

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-26: ext\_pmsssr bit assignments

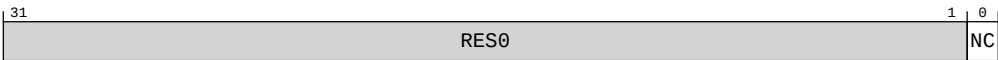


Table B-54: PMSSSR bit descriptions

| Bits   | Name | Description                                                                                                                                                    | Reset |
|--------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:1] | RES0 | Reserved                                                                                                                                                       | RES0  |
| [0]    | NC   | No capture. Indicates whether the PMU counters have been captured.<br><br><b>0b0</b><br>PMU counters captured.<br><br><b>0b1</b><br>PMU counters not captured. | x     |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PMU       | 0x610  | None  |

This interface is accessible as follows:

RO

B.3.5 PMCCNTSR, PMU Cycle Counter Snapshot Register

Captured copy of PMCCNTR\_ELO. Once captured, the value in PMCCNTSR is unaffected by writes to PMCCNTR\_ELO and PMCR\_ELO.C.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

PMU

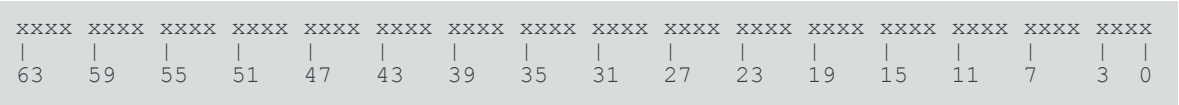
Register offset

0x618

Access type

RO

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-27: ext\_pmcntsr bit assignments

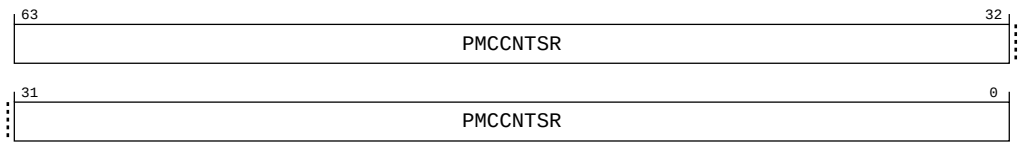


Table B-56: PMCCNTSR bit descriptions

| Bits   | Name     | Description                              | Reset    |
|--------|----------|------------------------------------------|----------|
| [63:0] | PMCCNTSR | PMCCNTR_ELO sample. Sampled cycle count. | 64 { x } |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PMU       | 0x618  | None  |

This interface is accessible as follows:

RO

B.3.6 PMCFGR, Performance Monitors Configuration Register

Contains PMU-specific configuration data.

Configurations

PMCFGR is in the Core power domain.

Attributes

Width

32

Component

PMU

Register offset

0xE00

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0000 | xxxx | xx1x | 0000 | 0111 | 1111 | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure B-28: pmu\_pmcfr bit assignments

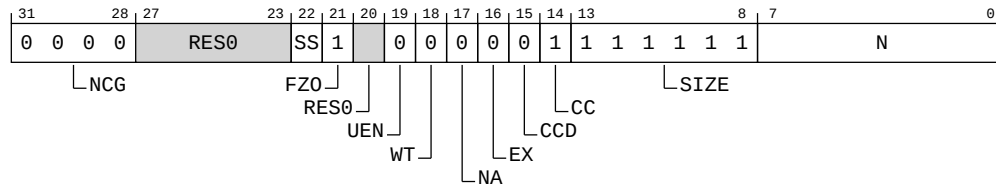


Table B-58: PMCFGR bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                | Reset                                                                     |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| [31:28] | NCG  | Defines the number of counter groups implemented, minus one.<br><br>This field reads-as-zero.<br><b>0b0000</b>                                                                                                                                                                                             | 0b0000                                                                    |
| [27:23] | RES0 | Reserved                                                                                                                                                                                                                                                                                                   | RES0                                                                      |
| [22]    | SS   | Snapshot supported.<br><b>0b0</b><br>Snapshot mechanism not supported. The locations 0x600-0x7FC and 0xE30-0xE3C are <b>IMPLEMENTATION DEFINED</b> .<br><b>0b1</b><br>Snapshot mechanism supported.<br><br>Locations 0x600-0x7FC and 0xE30-0xE3C contain <b>IMPLEMENTATION DEFINED</b> snapshot registers. | The reset values can be the following: 0b0, 0b1, respective to the value. |
| [21]    | FZO  | Freeze-on-overflow supported. Defined values are:<br><b>0b1</b><br>Freeze-on-overflow mechanism is supported. PMU.PMCR_ELO.FZO is RW.                                                                                                                                                                      | 0b1                                                                       |
| [20]    | RES0 | Reserved                                                                                                                                                                                                                                                                                                   | RES0                                                                      |
| [19]    | UEN  | User-mode Enable Register supported. AArch64-PMUSERENR_ELO is not visible in the external debug interface, so this bit is <b>RAZ</b> .<br><b>0b0</b>                                                                                                                                                       | 0b0                                                                       |
| [18]    | WT   | This feature is not supported, so this bit is <b>RAZ</b> .<br><b>0b0</b>                                                                                                                                                                                                                                   | 0b0                                                                       |
| [17]    | NA   | This feature is not supported, so this bit is <b>RAZ</b> .<br><b>0b0</b>                                                                                                                                                                                                                                   | 0b0                                                                       |
| [16]    | EX   | Export supported.<br><b>0b0</b><br>PMU.PMCR_ELO.X is <b>RES0</b> .                                                                                                                                                                                                                                         | 0b0                                                                       |
| [15]    | CCD  | Cycle counter has prescale.<br><b>0b0</b><br>PMU.PMCR_ELO.D is <b>RES0</b> .                                                                                                                                                                                                                               | 0b0                                                                       |

| Bits   | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                | Reset                                                                                      |
|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| [14]   | CC   | Dedicated cycle counter (counter 31) supported.<br><b>0b1</b>                                                                                                                                                                                                                                                                                                                                              | 0b1                                                                                        |
| [13:8] | SIZE | Size of counters, minus one. This field defines the size of the largest counter implemented by the Performance Monitors Unit.<br><br>From Armv8, the largest counter is 64-bits, so the value of this field is 0b111111.<br><br>This field is used by software to determine the spacing of the counters in the memory-map. From Armv8, the counters are a doubleword-aligned addresses.<br><b>0b111111</b> | 0b111111                                                                                   |
| [7:0]  | N    | Number of counters implemented in addition to the cycle counter, ext-PMCCNTR_ELO.<br><b>0b00000110</b><br>6 PMU counters<br><b>0b00011111</b><br>31 PMU counters<br><br>Must be configured to either 0x06 or 0x1f                                                                                                                                                                                          | The reset values can be the following:<br>0b00000110, 0b00011111, respective to the value. |

## Access

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

## Accessibility

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE00  | PMCFGR   | 31:0  |

This interface is accessible as follows:

**When DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalPMUAccess()**

ERROR

**Otherwise**

RO

### B.3.7 PMCR\_EL0, Performance Monitors Control Register

Provides details of the Performance Monitors implementation, including the number of counters implemented, and configures and controls the counters.

#### Configurations

PMCR\_EL0 is in the Core power domain.

#### Attributes

##### Width

32

##### Component

PMU

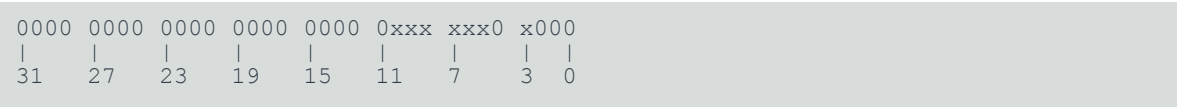
##### Register offset

0xE04

##### Access type

See bit descriptions

##### Reset value



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-29: pmu\_pmcr\_el0 bit assignments

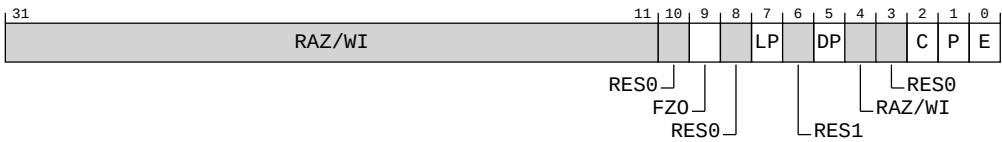


Table B-60: PMCR\_EL0 bit descriptions

| Bits    | Name   | Description | Reset  |
|---------|--------|-------------|--------|
| [31:11] | RAZ/WI | Reserved    | RAZ/WI |
| [10]    | RES0   | Reserved    | RES0   |



| Bits | Name       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset      |
|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [9]  | FZO        | <p>Freeze-on-overflow. Stop event counters on overflow.</p> <p>In the description of this field:</p> <ul style="list-style-type: none"> <li>If EL2 is implemented and is using AArch64, PMN is AArch64-MDCR_EL2.HPMN.</li> <li>If EL2 is not implemented, PMN is PMCR_ELO.N.</li> </ul> <p><b>0b0</b><br/>Do not freeze on overflow.</p> <p><b>0b1</b><br/>Affected counters do not count when PMU.PMOVSLR_ELO[(PMN-1):0] is nonzero.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | x          |
| [8]  | RES0       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RES0       |
| [7]  | LP         | <p>Long event counter enable. Determines when unsigned overflow is recorded by a counter overflow bit.</p> <p><b>0b1</b><br/>Event counter overflow on increment that causes unsigned overflow of PMU.PMEVCNTR&lt;n&gt;_ELO[63:0].</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | x          |
| [6]  | RES1       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RES1       |
| [5]  | DP         | <p>Disable cycle counter when event counting is prohibited. The possible values of this bit are:</p> <p><b>0b0</b><br/>Cycle counting by PMU.PMCCNTR_ELO is not affected by this mechanism.</p> <p><b>0b1</b><br/>Cycle counting by PMU.PMCCNTR_ELO is disabled in prohibited regions and when event counting is frozen:</p> <ul style="list-style-type: none"> <li>If FEAT_PMUv3p1 is implemented, EL2 is implemented, and AArch64-MDCR_EL2.HPMD is 1, then cycle counting by PMU.PMCCNTR_ELO is disabled at EL2.</li> <li>If FEAT_PMUv3p7 is implemented, EL3 is implemented and using AArch64, and AArch64-MDCR_EL3.MPMX is 1, then cycle counting by PMU.PMCCNTR_ELO is disabled at EL3.</li> <li>If FEAT_PMUv3p7 is implemented and event counting is frozen by PMCR_ELO.FZO, then cycle counting by PMU.PMCCNTR_ELO is disabled.</li> <li>If EL3 is implemented, AArch64-MDCR_EL3.SPME is 0, and either FEAT_PMUv3p7 is not implemented or AArch64-MDCR_EL3.MPMX is 0, then cycle counting by PMU.PMCCNTR_ELO is disabled at EL3 and in Secure state.</li> </ul> <p>If AArch64-MDCR_EL2.HPMN is not 0, this is when event counting by event counters in the range [0..(AArch64-MDCR_EL2.HPMN-1)] is prohibited or frozen.</p> | x          |
| [4]  | RAZ/<br>WI | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RAZ/<br>WI |
| [3]  | RES0       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RES0       |
| [2]  | C          | <p>Cycle counter reset. The effects of writing to this bit are:</p> <p><b>0b1</b><br/>Reset PMU.PMCCNTR_ELO to zero.</p> <p>Access to this field is: WO/RAZ</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b0        |
| [1]  | P          | <p>Event counter reset. The effects of writing to this bit are:</p> <p><b>0b1</b><br/>Reset all event counters, not including PMU.PMCCNTR_ELO, to zero.</p> <p>Access to this field is: WO/RAZ</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0b0        |

| Bits | Name | Description                                                                                                                                                                    | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | E    | Enable<br><br><b>0b1</b><br><br>PMU.PMCCNTR_ELO and event counters PMU.PMEVCNTR<n>_ELO, where n is in the range of affected event counters, are enabled by PMU.PMCNTENSET_ELO. | 0b0   |

### Access

SoftwareLockStatus() depends on the type of access attempted and AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

### Accessibility

SoftwareLockStatus() depends on the type of access attempted and AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE04  | PMCR_ELO | None  |

This interface is accessible as follows:

**When DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalPMUAccess()**

ERROR

**When SoftwareLockStatus()**

RO

**Otherwise**

RW

## B.3.8 PMCEID0, Performance Monitors Common Event Identification register 0

Defines which Common architectural events and Common microarchitectural events are implemented, or counted, using PMU events in the range 0x0000 to 0x001F.

For more information about the Common events and the use of the PMCEIDn registers, see *The PMU event number space and common events* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



This view of the register was previously called PMCEID0\_ELO.

### Configurations

PMCEID0 is in the Core power domain.

Attributes

Width

32

Component

PMU

Register offset

0xE20

Access type

See bit descriptions

Reset value

0111 1111 1111 1111 0110 1111 0011 1111

Bit descriptions

Figure B-30: pmu\_pmceid0 bit assignments

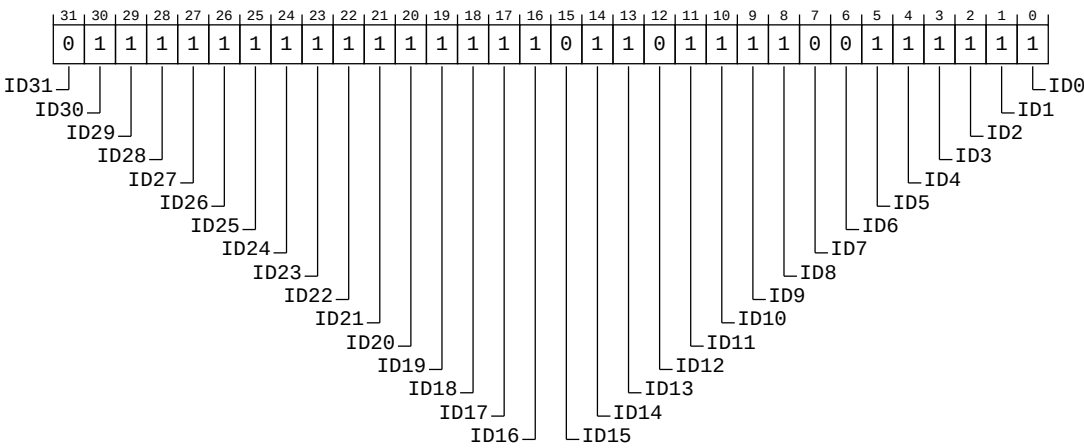


Table B-62: PMCEID0 bit descriptions

| Bits | Name | Description                                                                                                                      | Reset |
|------|------|----------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | ID31 | ID31 corresponds to common event (0x1f) L1D_CACHE_ALLOCATE<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [30] | ID30 | ID30 corresponds to common event (0x1e) CHAIN<br><b>0b1</b><br>The Common event is implemented.                                  | 0b1   |
| [29] | ID29 | ID29 corresponds to common event (0x1d) BUS_CYCLES<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |

| Bits | Name | Description                                                                                                                         | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------|-------|
| [28] | ID28 | ID28 corresponds to common event (0x1c) TTBR_WRITE_RETIRED<br><b>0b1</b><br>The Common event is implemented.                        | 0b1   |
| [27] | ID27 | ID27 corresponds to common event (0x1b) INST_SPEC<br><b>0b1</b><br>The Common event is implemented.                                 | 0b1   |
| [26] | ID26 | ID26 corresponds to common event (0x1a) MEMORY_ERROR<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [25] | ID25 | ID25 corresponds to common event (0x19) BUS_ACCESS<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |
| [24] | ID24 | ID24 corresponds to common event (0x18) L2D_CACHE_WB<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [23] | ID23 | ID23 corresponds to common event (0x17) L2D_CACHE_REFILL<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [22] | ID22 | ID22 corresponds to common event (0x16) L2D_CACHE<br><b>0b1</b><br>The Common event is implemented.                                 | 0b1   |
| [21] | ID21 | ID21 corresponds to common event (0x15) L1D_CACHE_WB<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [20] | ID20 | ID20 corresponds to common event (0x14) L1I_CACHE<br><b>0b1</b><br>The Common event is implemented.                                 | 0b1   |
| [19] | ID19 | ID19 corresponds to common event (0x13) MEM_ACCESS<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |
| [18] | ID18 | ID18 corresponds to common event (0x12) BR_PRED<br><b>0b1</b><br>The Common event is implemented.                                   | 0b1   |
| [17] | ID17 | ID17 corresponds to common event (0x11) CPU_CYCLES<br><b>0b1</b><br>The Common event is implemented.                                | 0b1   |
| [16] | ID16 | ID16 corresponds to common event (0x10) BR_MIS_PRED<br><b>0b1</b><br>The Common event is implemented.                               | 0b1   |
| [15] | ID15 | ID15 corresponds to common event (0xf) UNALIGNED_LDST_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |

| Bits | Name | Description                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| [14] | ID14 | ID14 corresponds to common event (0xe) BR_RETURN_RETIRED<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [13] | ID13 | ID13 corresponds to common event (0xd) BR_IMMED_RETIRED<br><b>0b1</b><br>The Common event is implemented.                     | 0b1   |
| [12] | ID12 | ID12 corresponds to common event (0xc) PC_WRITE_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [11] | ID11 | ID11 corresponds to common event (0xb) CID_WRITE_RETIRED<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [10] | ID10 | ID10 corresponds to common event (0xa) EXC_RETURN<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [9]  | ID9  | ID9 corresponds to common event (0x9) EXC_TAKEN<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [8]  | ID8  | ID8 corresponds to common event (0x8) INST_RETIRED<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [7]  | ID7  | ID7 corresponds to common event (0x7) ST_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [6]  | ID6  | ID6 corresponds to common event (0x6) LD_RETIRED<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [5]  | ID5  | ID5 corresponds to common event (0x5) L1D_TLB_REFILL<br><b>0b1</b><br>The Common event is implemented.                        | 0b1   |
| [4]  | ID4  | ID4 corresponds to common event (0x4) L1D_CACHE<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [3]  | ID3  | ID3 corresponds to common event (0x3) L1D_CACHE_REFILL<br><b>0b1</b><br>The Common event is implemented.                      | 0b1   |
| [2]  | ID2  | ID2 corresponds to common event (0x2) L1I_TLB_REFILL<br><b>0b1</b><br>The Common event is implemented.                        | 0b1   |
| [1]  | ID1  | ID1 corresponds to common event (0x1) L1I_CACHE_REFILL<br><b>0b1</b><br>The Common event is implemented.                      | 0b1   |

| Bits | Name | Description                                                                                         | Reset |
|------|------|-----------------------------------------------------------------------------------------------------|-------|
| [0]  | ID0  | ID0 corresponds to common event (0x0) SW_INCR<br><br><b>0b1</b><br>The Common event is implemented. | 0b1   |

**Access**

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

**Accessibility**

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE20  | PMCEID0  | None  |

This interface is accessible as follows:

**When** DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalIPMUAccess()  
ERROR

**Otherwise**  
RO

B.3.9 PMCEID1, Performance Monitors Common Event Identification register 1

Defines which Common architectural events and Common microarchitectural events are implemented, or counted, using PMU events in the range 0x020 to 0x03F.

For more information about the Common events and the use of the PMCEIDn registers, see *The PMU event number space and common events* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



This view of the register was previously called PMCEID1\_EL0.

**Configurations**

PMCEID1 is in the Core power domain.

**Attributes**

**Width**  
32

**Component**

PMU

**Register offset**

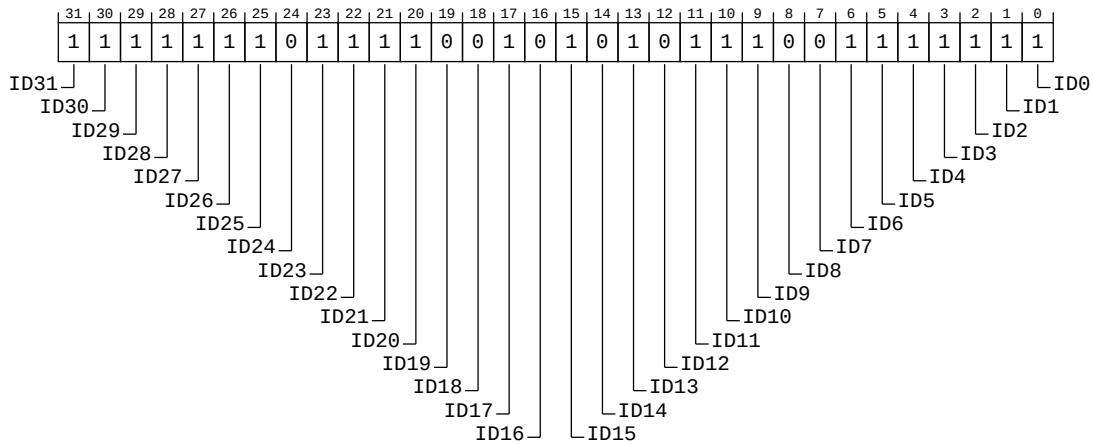
0xE24

**Access type**

See bit descriptions

**Reset value**

1111 1110 1111 0010 1010 1110 0111 1111

**Bit descriptions****Figure B-31: pmu\_pmceid1 bit assignments****Table B-64: PMCEID1 bit descriptions**

| Bits | Name | Description                                                                                                   | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------|-------|
| [31] | ID31 | ID31 corresponds to common event (0x3f) STALL_SLOT<br><b>0b1</b><br>The Common event is implemented.          | 0b1   |
| [30] | ID30 | ID30 corresponds to common event (0x3e) STALL_SLOT_FRONTEND<br><b>0b1</b><br>The Common event is implemented. | 0b1   |
| [29] | ID29 | ID29 corresponds to common event (0x3d) STALL_SLOT_BACKEND<br><b>0b1</b><br>The Common event is implemented.  | 0b1   |
| [28] | ID28 | ID28 corresponds to common event (0x3c) STALL<br><b>0b1</b><br>The Common event is implemented.               | 0b1   |
| [27] | ID27 | ID27 corresponds to common event (0x3b) OP_SPEC<br><b>0b1</b><br>The Common event is implemented.             | 0b1   |

| Bits | Name | Description                                                                                                                    | Reset |
|------|------|--------------------------------------------------------------------------------------------------------------------------------|-------|
| [26] | ID26 | ID26 corresponds to common event (0x3a) OP_RETIRED<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [25] | ID25 | ID25 corresponds to common event (0x39) L1D_CACHE_LMISS_RD<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [24] | ID24 | ID24 corresponds to common event (0x38) REMOTE_ACCESS_RD<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [23] | ID23 | ID23 corresponds to common event (0x37) LL_CACHE_MISS_RD<br><b>0b1</b><br>The Common event is implemented.                     | 0b1   |
| [22] | ID22 | ID22 corresponds to common event (0x36) LL_CACHE_RD<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [21] | ID21 | ID21 corresponds to common event (0x35) ITLB_WLK<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [20] | ID20 | ID20 corresponds to common event (0x34) DTLB_WLK<br><b>0b1</b><br>The Common event is implemented.                             | 0b1   |
| [19] | ID19 | ID19 corresponds to a Reserved Event event (0x33)<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [18] | ID18 | ID18 corresponds to a Reserved Event event (0x32)<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [17] | ID17 | ID17 corresponds to common event (0x31) REMOTE_ACCESS<br><b>0b1</b><br>The Common event is implemented.                        | 0b1   |
| [16] | ID16 | ID16 corresponds to common event (0x30) L2I_TLB<br><b>0b0</b><br>The Common event is not implemented, or not counted.          | 0b0   |
| [15] | ID15 | ID15 corresponds to common event (0x2f) L2TLB_REQ<br><b>0b1</b><br>The Common event is implemented.                            | 0b1   |
| [14] | ID14 | ID14 corresponds to common event (0x2e) L2I_TLB_REFILL<br><b>0b0</b><br>The Common event is not implemented, or not counted.   | 0b0   |
| [13] | ID13 | ID13 corresponds to common event (0x2d) L2TLB_REFILL<br><b>0b1</b><br>The Common event is implemented.                         | 0b1   |



| Bits | Name | Description                                                                                                                   | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| [12] | ID12 | ID12 corresponds to common event (0x2c) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [11] | ID11 | ID11 corresponds to common event (0x2b) L3D_CACHE<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [10] | ID10 | ID10 corresponds to common event (0x2a) L3D_CACHE_REFILL<br><b>0b1</b><br>The Common event is implemented.                    | 0b1   |
| [9]  | ID9  | ID9 corresponds to common event (0x29) L3D_CACHE_ALLOCATE<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [8]  | ID8  | ID8 corresponds to common event (0x28) L2I_CACHE_REFILL<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [7]  | ID7  | ID7 corresponds to common event (0x27) L2I_CACHE<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [6]  | ID6  | ID6 corresponds to common event (0x26) L1I_TLB<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [5]  | ID5  | ID5 corresponds to common event (0x25) L1D_TLB<br><b>0b1</b><br>The Common event is implemented.                              | 0b1   |
| [4]  | ID4  | ID4 corresponds to common event (0x24) STALL_BACKEND<br><b>0b1</b><br>The Common event is implemented.                        | 0b1   |
| [3]  | ID3  | ID3 corresponds to common event (0x23) STALL_FRONTEND<br><b>0b1</b><br>The Common event is implemented.                       | 0b1   |
| [2]  | ID2  | ID2 corresponds to common event (0x22) BR_MIS_PRED_RETIRED<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [1]  | ID1  | ID1 corresponds to common event (0x21) BR_RETIRED<br><b>0b1</b><br>The Common event is implemented.                           | 0b1   |
| [0]  | ID0  | ID0 corresponds to common event (0x20) L2D_CACHE_ALLOCATE<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |

## Access

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

## Accessibility

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE24  | PMCEID1  | None  |

This interface is accessible as follows:

**When DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalPMUAccess()**

ERROR

**Otherwise**

RO

## B.3.10 PMCEID2, Performance Monitors Common Event Identification register 2

Defines which Common architectural events and Common microarchitectural events are implemented, or counted, using PMU events in the range 0x4000 to 0x401F.

For more information about the Common events and the use of the PMCEIDn registers, see *The PMU event number space and common events* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

## Configurations

PMCEID2 is in the Core power domain.

## Attributes

### Width

32

### Component

PMU

### Register offset

0xE28

### Access type

See bit descriptions

### Reset value

0000 1111 0000 1111 0001 1010 0111 1111

## Bit descriptions

**Figure B-32: pmu\_pmceid2 bit assignments**

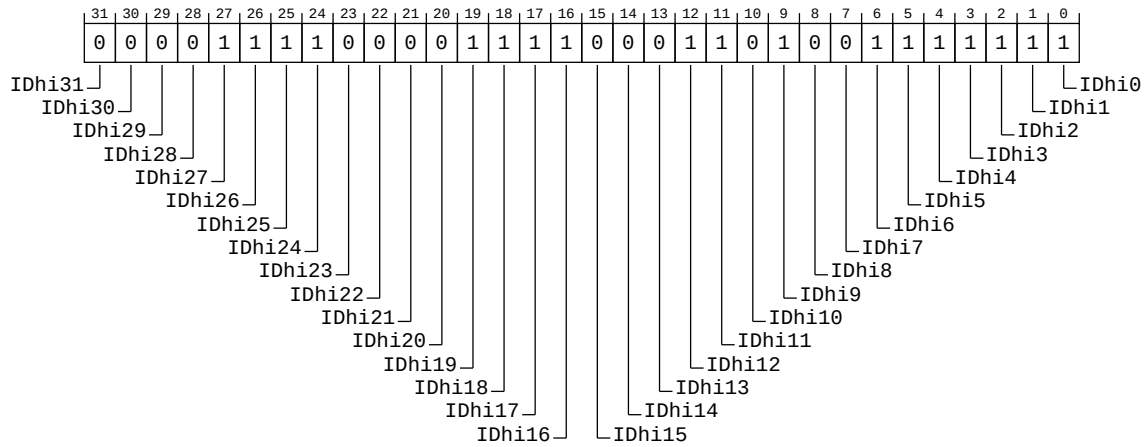


**Table B-66: PMCEID2 bit descriptions**

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | IDHi31 | IDHi31 corresponds to a Reserved Event event (0x401f)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [30] | IDHi30 | IDHi30 corresponds to a Reserved Event event (0x401e)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [29] | IDHi29 | IDHi29 corresponds to a Reserved Event event (0x401d)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [28] | IDHi28 | IDHi28 corresponds to a Reserved Event event (0x401c)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [27] | IDHi27 | IDHi27 corresponds to common event (0x401b) CTI_TRIGOUT7<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [26] | IDHi26 | IDHi26 corresponds to common event (0x401a) CTI_TRIGOUT6<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [25] | IDHi25 | IDHi25 corresponds to common event (0x4019) CTI_TRIGOUT5<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [24] | IDHi24 | IDHi24 corresponds to common event (0x4018) CTI_TRIGOUT4<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |

| Bits | Name   | Description                                                                                                                       | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------|-------|
| [23] | IDhi23 | IDhi23 corresponds to a Reserved Event event (0x4017)<br><b>0b0</b><br>The Common event is not implemented, or not counted.       | 0b0   |
| [22] | IDhi22 | IDhi22 corresponds to a Reserved Event event (0x4016)<br><b>0b0</b><br>The Common event is not implemented, or not counted.       | 0b0   |
| [21] | IDhi21 | IDhi21 corresponds to a Reserved Event event (0x4015)<br><b>0b0</b><br>The Common event is not implemented, or not counted.       | 0b0   |
| [20] | IDhi20 | IDhi20 corresponds to a Reserved Event event (0x4014)<br><b>0b0</b><br>The Common event is not implemented, or not counted.       | 0b0   |
| [19] | IDhi19 | IDhi19 corresponds to common event (0x4013) TRCEXTOUT3<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [18] | IDhi18 | IDhi18 corresponds to common event (0x4012) TRCEXTOUT2<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [17] | IDhi17 | IDhi17 corresponds to common event (0x4011) TRCEXTOUT1<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [16] | IDhi16 | IDhi16 corresponds to common event (0x4010) TRCEXTOUT0<br><b>0b1</b><br>The Common event is implemented.                          | 0b1   |
| [15] | IDhi15 | IDhi15 corresponds to common event (0x400f) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [14] | IDhi14 | IDhi14 corresponds to common event (0x400e) TRB_TRIG<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [13] | IDhi13 | IDhi13 corresponds to common event (0x400d) PMU_OVFS<br><b>0b0</b><br>The Common event is not implemented, or not counted.        | 0b0   |
| [12] | IDhi12 | IDhi12 corresponds to common event (0x400c) TRB_WRAP<br><b>0b1</b><br>The Common event is implemented.                            | 0b1   |
| [11] | IDhi11 | IDhi11 corresponds to common event (0x400b) L3D_CACHE_LMISS_RD<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [10] | IDhi10 | IDhi10 corresponds to common event (0x400a) L2I_CACHE_LMISS<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |

| Bits | Name  | Description                                                                                                               | Reset |
|------|-------|---------------------------------------------------------------------------------------------------------------------------|-------|
| [9]  | IDhi9 | IDhi9 corresponds to common event (0x4009) L2D_CACHE_LMISS_RD<br><b>0b1</b><br>The Common event is implemented.           | 0b1   |
| [8]  | IDhi8 | IDhi8 corresponds to common event (0x4008) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [7]  | IDhi7 | IDhi7 corresponds to common event (0x4007) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [6]  | IDhi6 | IDhi6 corresponds to common event (0x4006) L1I_CACHE_LMISS<br><b>0b1</b><br>The Common event is implemented.              | 0b1   |
| [5]  | IDhi5 | IDhi5 corresponds to common event (0x4005) STALL_BACKEND_MEM<br><b>0b1</b><br>The Common event is implemented.            | 0b1   |
| [4]  | IDhi4 | IDhi4 corresponds to common event (0x4004) CNT_CYCLES<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |
| [3]  | IDhi3 | IDhi3 corresponds to common event (0x4003) SAMPLE_COLLISION<br><b>0b1</b><br>The Common event is implemented.             | 0b1   |
| [2]  | IDhi2 | IDhi2 corresponds to common event (0x4002) SAMPLE_FILTRATE<br><b>0b1</b><br>The Common event is implemented.              | 0b1   |
| [1]  | IDhi1 | IDhi1 corresponds to common event (0x4001) SAMPLE_FEED<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [0]  | IDhi0 | IDhi0 corresponds to common event (0x4000) SAMPLE_POP<br><b>0b1</b><br>The Common event is implemented.                   | 0b1   |

## Access

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

## Accessibility

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE28  | PMCEID2  | None  |

This interface is accessible as follows:

**When DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalPMUAccess()**

ERROR

**Otherwise**

RO

### B.3.11 PMCEID3, Performance Monitors Common Event Identification register 3

Defines which Common architectural events and Common microarchitectural events are implemented, or counted, using PMU events in the range 0x4020 to 0x403F.

For more information about the Common events and the use of the PMCEIDn registers, see *The PMU event number space and common events* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

PMCEID3 is in the Core power domain.

#### Attributes

##### Width

32

##### Component

PMU

##### Register offset

0xE2C

##### Access type

See bit descriptions

##### Reset value

0000 0000 0000 0000 0000 0000 0111 0111

## Bit descriptions

**Figure B-33: pmu\_pmceid3 bit assignments**

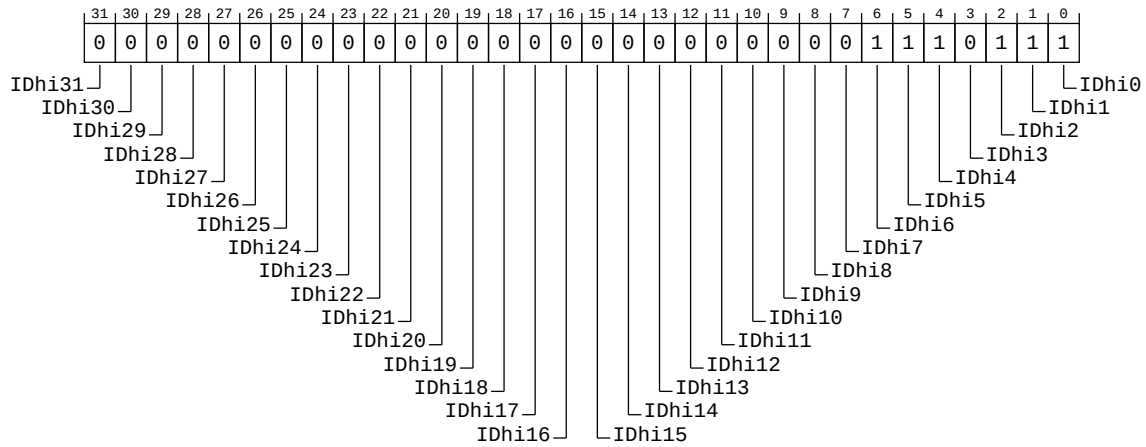


**Table B-68: PMCEID3 bit descriptions**

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | IDhi31 | IDhi31 corresponds to a Reserved Event event (0x403f)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [30] | IDhi30 | IDhi30 corresponds to a Reserved Event event (0x403e)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [29] | IDhi29 | IDhi29 corresponds to a Reserved Event event (0x403d)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [28] | IDhi28 | IDhi28 corresponds to a Reserved Event event (0x403c)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [27] | IDhi27 | IDhi27 corresponds to a Reserved Event event (0x403b)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [26] | IDhi26 | IDhi26 corresponds to a Reserved Event event (0x403a)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [25] | IDhi25 | IDhi25 corresponds to a Reserved Event event (0x4039)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [24] | IDhi24 | IDhi24 corresponds to a Reserved Event event (0x4038)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |

| Bits | Name   | Description                                                                                                                 | Reset |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| [23] | IDHi23 | IDHi23 corresponds to a Reserved Event event (0x4037)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [22] | IDHi22 | IDHi22 corresponds to a Reserved Event event (0x4036)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [21] | IDHi21 | IDHi21 corresponds to a Reserved Event event (0x4035)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [20] | IDHi20 | IDHi20 corresponds to a Reserved Event event (0x4034)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [19] | IDHi19 | IDHi19 corresponds to a Reserved Event event (0x4033)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [18] | IDHi18 | IDHi18 corresponds to a Reserved Event event (0x4032)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [17] | IDHi17 | IDHi17 corresponds to a Reserved Event event (0x4031)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [16] | IDHi16 | IDHi16 corresponds to a Reserved Event event (0x4030)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [15] | IDHi15 | IDHi15 corresponds to a Reserved Event event (0x402f)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [14] | IDHi14 | IDHi14 corresponds to a Reserved Event event (0x402e)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [13] | IDHi13 | IDHi13 corresponds to a Reserved Event event (0x402d)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [12] | IDHi12 | IDHi12 corresponds to a Reserved Event event (0x402c)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [11] | IDHi11 | IDHi11 corresponds to a Reserved Event event (0x402b)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [10] | IDHi10 | IDHi10 corresponds to a Reserved Event event (0x402a)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |



| Bits | Name  | Description                                                                                                                | Reset |
|------|-------|----------------------------------------------------------------------------------------------------------------------------|-------|
| [9]  | IDhi9 | IDhi9 corresponds to a Reserved Event event (0x4029)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [8]  | IDhi8 | IDhi8 corresponds to a Reserved Event event (0x4028)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [7]  | IDhi7 | IDhi7 corresponds to a Reserved Event event (0x4027)<br><b>0b0</b><br>The Common event is not implemented, or not counted. | 0b0   |
| [6]  | IDhi6 | IDhi6 corresponds to common event (0x4026) MEM_ACCESS_CHECKED_WR<br><b>0b1</b><br>The Common event is implemented.         | 0b1   |
| [5]  | IDhi5 | IDhi5 corresponds to common event (0x4025) MEM_ACCESS_CHECKED_RD<br><b>0b1</b><br>The Common event is implemented.         | 0b1   |
| [4]  | IDhi4 | IDhi4 corresponds to common event (0x4024) MEM_ACCESS_CHECKED<br><b>0b1</b><br>The Common event is implemented.            | 0b1   |
| [3]  | IDhi3 | IDhi3 corresponds to common event (0x4023) Reserved<br><b>0b0</b><br>The Common event is not implemented, or not counted.  | 0b0   |
| [2]  | IDhi2 | IDhi2 corresponds to common event (0x4022) ST_ALIGN_LAT<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [1]  | IDhi1 | IDhi1 corresponds to common event (0x4021) LD_ALIGN_LAT<br><b>0b1</b><br>The Common event is implemented.                  | 0b1   |
| [0]  | IDhi0 | IDhi0 corresponds to common event (0x4020) LDST_ALIGN_LAT<br><b>0b1</b><br>The Common event is implemented.                | 0b1   |

## Access

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

## Accessibility

AllowExternalPMUAccess() has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE2C  | PMCEID3  | None  |

This interface is accessible as follows:

When DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalPMUAccess()  
ERROR  
Otherwise  
RO

B.3.12 PMSSCR, PMU Snapshot Capture Register

Provides a mechanism for software to initiate a sample.

Configurations  
This register is available in all configurations.

Attributes  
Width  
32  
Component  
PMU  
Register offset  
0xE30  
Access type  
RESERVEDW  
Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions  
Figure B-34: ext\_pmsscr bit assignments

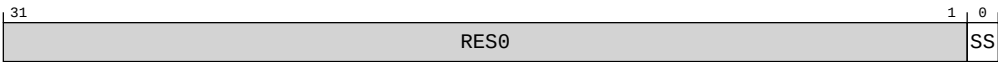


Table B-70: PMSSCR bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:1] | RES0 | Reserved    | RES0  |

| Bits | Name | Description                                                                                     | Reset |
|------|------|-------------------------------------------------------------------------------------------------|-------|
| [0]  | SS   | Capture now.<br><br><b>0b0</b><br>Ignored.<br><br><b>0b1</b><br>Initiate a capture immediately. | x     |

Accessibility

| Component | Offset | Range |
|-----------|--------|-------|
| PMU       | 0xE30  | None  |

This interface is accessible as follows:

WO

B.3.13 PMMIR, Performance Monitors Machine Identification Register

Describes Performance Monitors parameters specific to the implementation.

Configurations

PMMIR is in the Core power domain.

Attributes

Width

32

Component

PMU

Register offset

0xE40

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 1010 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-35: pmu\_pmmir bit assignments

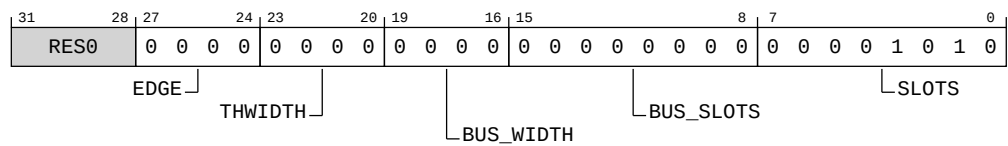


Table B-72: PMMIR bit descriptions

| Bits    | Name      | Description                                                                                                                                                                                                                                                                               | Reset  |
|---------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:28] | RES0      | Reserved                                                                                                                                                                                                                                                                                  | RES0   |
| [27:24] | EDGE      | PMU event edge detection. Indicates implementation of the FEAT_PMUv3_EDGE feature.<br><br>0b0000<br>FEAT_PMUv3_EDGE is not implemented.                                                                                                                                                   | 0b0000 |
| [23:20] | THWIDTH   | PMU.PMEVTYPER<n>_ELO.TH width. Indicates implementation of the FEAT_PMUv3_TH feature, and, if implemented, the size of the PMU.PMEVTYPER<n>_ELO.TH field.<br><br>0b0000<br>FEAT_PMUv3_TH is not implemented.                                                                              | 0b0000 |
| [19:16] | BUS_WIDTH | Bus width. Indicates the number of bytes each BUS_ACCESS event relates to. Encoded as Log <sub>2</sub> (number of bytes), plus one.<br><br>0b0000<br>The information is not available.                                                                                                    | 0b0000 |
| [15:8]  | BUS_SLOTS | Bus count. The largest value by which the BUS_ACCESS event might increment in a single BUS_CYCLES cycle.<br><br>0b00000000<br>The largest value by which the BUS_ACCESS event might increment in a single BUS_CYCLES cycle is 0                                                           | 0x00   |
| [7:0]   | SLOTS     | Operation width. The largest value by which the STALL_SLOT event might increment in a single cycle. If the STALL_SLOT event is not implemented, this field might read as zero.<br><br>0b00001010<br>The largest value by which the STALL_SLOT PMU event may increment in one cycle is 10. | 0x0A   |

Accessibility

If the Core power domain is off or in a low-power state, access on this interface returns an Error.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xE40  | PMMIR    | 31:0  |

This interface is accessible as follows:

```
When DoubleLockStatus() || !IsCorePowered() || OSLockStatus() || !AllowExternalPMUAccess()
    ERROR
Otherwise
    RO
```

B.3.14 PMDEVARCH, Performance Monitors Device Architecture register

Identifies the programmers' model architecture of the Performance Monitor component.

Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

Attributes

Width

32

Component

PMU

Register offset

0xFBC

Access type

See bit descriptions

Reset value

0100 0111 0111 0000 0010 1010 0001 0110

Bit descriptions

Figure B-36: pmu\_pmdevarch bit assignments

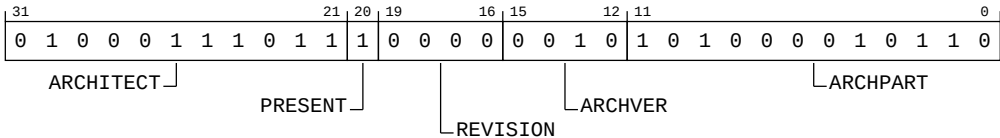


Table B-74: PMDEVARCH bit descriptions

| Bits    | Name      | Description                                                                                                                                                                                                                        | Reset         |
|---------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [31:21] | ARCHITECT | Defines the architecture of the component. For Performance Monitors, this is Arm Limited.<br><br>Bits [31:28] are the JEP106 continuation code, 0x4.<br><br>Bits [27:21] are the JEP106 ID code, 0x3B.<br><br><b>0b01000111011</b> | 0b01000111011 |
| [20]    | PRESENT   | Indicates that the DEVARCH is present.<br><br><b>0b1</b>                                                                                                                                                                           | 0b1           |

| Bits    | Name     | Description                                                                                                                                                                                                                   | Reset  |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [19:16] | REVISION | Defines the architecture revision. For architectures defined by Arm this is the minor revision.<br><br>For Performance Monitors, the revision defined by Armv8 is 0x0.<br><br>All other values are reserved.<br><b>0b0000</b> | 0b0000 |
| [15:12] | ARCHVER  | Architecture Version. Defines the architecture version of the component.<br><b>0b0010</b><br>Performance Monitors Extension version 3, PMUv3.                                                                                 | 0b0010 |
| [11:0]  | ARCHPART | Architecture Part. Defines the architecture of the component.<br><b>0b101000010110</b><br>Armv8-A PE performance monitors.                                                                                                    | 0xA16  |

### Accessibility

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| PMU       | 0xFBC  | PMDEVARCH | None  |

This interface is accessible as follows:

#### When !IsCorePowered()

ERROR

#### Otherwise

RO

## B.3.15 PMDEVID, Performance Monitors Device ID register

Provides information about features of the Performance Monitors implementation.

### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain.

If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required from Armv8.2 and in any implementation that includes FEAT\_PCSRv8p2. Otherwise, its location is RES0.



Note

Before Armv8.2, the PC Sample-based Profiling Extension can be implemented in the external debug register space, as indicated by the value of ext-EDDEVID.PCSample.

Attributes

Width

32

Component

PMU

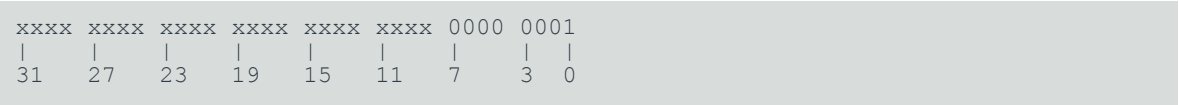
Register offset

0xFC8

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-37: pmu\_pmdevid bit assignments



Table B-76: PMDEVID bit descriptions

| Bits   | Name     | Description                                                                                                                                                                                                                               | Reset  |
|--------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0     | Reserved                                                                                                                                                                                                                                  | RES0   |
| [7:4]  | PMSS     | PMU Snapshot extension. Defined values are:<br><br><b>0b0000</b><br>PMU snapshot extension not implemented.                                                                                                                               | 0b0000 |
| [3:0]  | PCSample | Indicates the level of PC Sample-based Profiling support using Performance Monitors registers. Defined values are:<br><br><b>0b0001</b><br>PC Sample-based Profiling Extension is implemented in the Performance Monitors register space. | 0b0001 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFC8  | PMDEVID  | None  |

This interface is accessible as follows:

When `!IsCorePowered()`

ERROR

Otherwise

RO

B.3.16 PMDEVTYPE, Performance Monitors Device Type register

Indicates to a debugger that this component is part of a PE's performance monitor interface.

Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

Attributes

Width

32

Component

PMU

Register offset

0xFCC

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0001 | 0110 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-38: pmu\_pmdevtype bit assignments

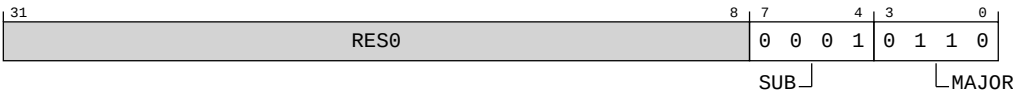




**Table B-78: PMDEVTYPE bit descriptions**

| Bits   | Name  | Description                                                                     | Reset  |
|--------|-------|---------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                        | RES0   |
| [7:4]  | SUB   | Subtype. Indicates this is a component within a PE.<br><b>0b0001</b>            | 0b0001 |
| [3:0]  | MAJOR | Major type. Indicates this is a performance monitor component.<br><b>0b0110</b> | 0b0110 |

### Accessibility

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| PMU       | 0xFCC  | PMDEVTYPE | None  |

This interface is accessible as follows:

**When !IsCorePowered()**

ERROR

**Otherwise**

RO

## B.3.17 PMPIDR4, Performance Monitors Peripheral Identification Register 4

Provides information to identify a Performance Monitor component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

### Attributes

**Width**

32

**Component**

PMU

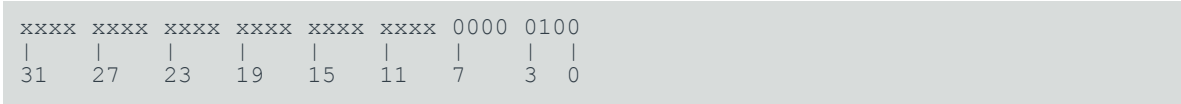
**Register offset**

0xFD0

**Access type**

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-39: pmu\_pmpidr4 bit assignments



Table B-80: PMPIDR4 bit descriptions

| Bits   | Name  | Description                                                                                                                                                                       | Reset  |
|--------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                                                                          | RES0   |
| [7:4]  | SIZE  | 4KB count.<br>0b0000                                                                                                                                                              | 0b0000 |
| [3:0]  | DES_2 | Designer, JEP106 continuation code, least significant nibble. For Arm Limited, this field is 0b0100.<br>0b0100<br>Arm Limited. This is bits[3:0] of the JEP106 continuation code. | 0b0100 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFD0  | PMPIDR4  | None  |

This interface is accessible as follows:

When !IsCorePowered()

ERROR

Otherwise

RO

### B.3.18 PMPIDR0, Performance Monitors Peripheral Identification Register 0

Provides information to identify a Performance Monitor component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

#### Attributes

##### Width

32

##### Component

PMU

##### Register offset

0xFE0

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1000 | 0101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-40: pmu\_pmpidr0 bit assignments



**Table B-82: PMPIDR0 bit descriptions**

| Bits   | Name   | Description                                                                                                   | Reset |
|--------|--------|---------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0   | Reserved                                                                                                      | RES0  |
| [7:0]  | PART_0 | Part number, least significant byte.<br><br><b>0b10000101</b><br>Least significant byte of the PMU unit part. | 0x85  |

**Accessibility**

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFE0  | PMPIDR0  | None  |

This interface is accessible as follows:

**When !IsCorePowered()**

ERROR

**Otherwise**

RO

### B.3.19 PMPIDR1, Performance Monitors Peripheral Identification Register 1

Provides information to identify a Performance Monitor component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Configurations**

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

**Attributes****Width**

32

**Component**

PMU

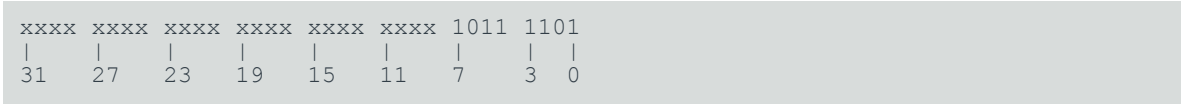
**Register offset**

0xFE4

**Access type**

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-41: pmu\_pmpidr1 bit assignments

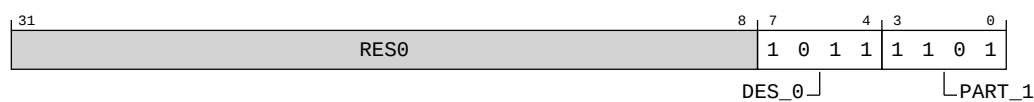


Table B-84: PMPIDR1 bit descriptions

| Bits   | Name   | Description                                                                                                                                                                        | Reset  |
|--------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                                                                                                           | RES0   |
| [7:4]  | DES_0  | Designer, least significant nibble of JEP106 ID code. For Arm Limited, this field is 0b1011.<br><br>0b1011<br>Arm Limited. This is the least significant nibble of JEP106 ID code. | 0b1011 |
| [3:0]  | PART_1 | Part number, most significant nibble.<br><br>0b1101<br>Part number, most significant nibble.                                                                                       | 0b1101 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFE4  | PMPIDR1  | None  |

This interface is accessible as follows:

When !IsCorePowered()

ERROR

Otherwise

RO

## B.3.20 PMPIDR2, Performance Monitors Peripheral Identification Register 2

Provides information to identify a Performance Monitor component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

### Attributes

#### Width

32

#### Component

PMU

#### Register offset

0xFE8

#### Access type

See bit descriptions

#### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

### Bit descriptions

Figure B-42: pmu\_pmpidr2 bit assignments

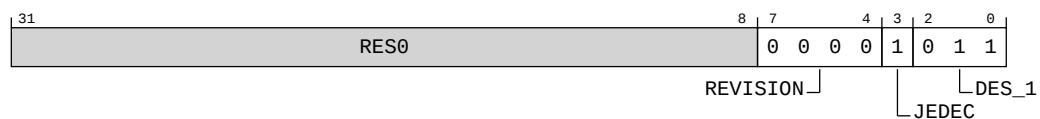


**Table B-86: PMPIDR2 bit descriptions**

| Bits   | Name     | Description                                                                                                                                                           | Reset  |
|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0     | Reserved                                                                                                                                                              | RES0   |
| [7:4]  | REVISION | Part major revision. Parts can also use this field to extend Part number to 16-bits.<br><br><b>0b0000</b><br>r0p2                                                     | 0b0000 |
| [3]    | JEDEC    | Indicates a JEP106 identity code is used.<br><br><b>0b1</b>                                                                                                           | 0b1    |
| [2:0]  | DES_1    | Designer, most significant bits of JEP106 ID code. For Arm Limited, this field is 0b011.<br><br><b>0b011</b><br>Arm Limited. This is bits[6:4] of the JEP106 ID code. | 0b011  |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFE8  | PMPIDR2  | None  |

This interface is accessible as follows:

**When !IsCorePowered()**

ERROR

**Otherwise**

RO

## B.3.21 PMPIDR3, Performance Monitors Peripheral Identification Register 3

Provides information to identify a Performance Monitor component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

### Attributes

**Width**

32

**Component**

PMU

Register offset

0xFEC

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-43: pmu\_pmpidr3 bit assignments



Table B-88: PMPIDR3 bit descriptions

| Bits   | Name   | Description                                                                                                                                                   | Reset  |
|--------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                                                                                      | RES0   |
| [7:4]  | REVAND | Part minor revision. Parts using PMU.PMPIDR2.REVISION as an extension to the Part number must use this field as a major revision number.<br>0b0010            | 0b0010 |
| [3:0]  | CMOD   | Customer modified. Indicates someone other than the Designer has modified the component.<br>0b0000<br>The component is not modified from the original design. | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFEC  | PMPIDR3  | None  |

This interface is accessible as follows:

When !IsCorePowered()

ERROR

Otherwise

RO



### B.3.22 PMCIDR0, Performance Monitors Component Identification Register 0

Provides information to identify a Performance Monitor component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

#### Attributes

##### Width

32

##### Component

PMU

##### Register offset

0xFF0

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-44: pmu\_pmcidr0 bit assignments

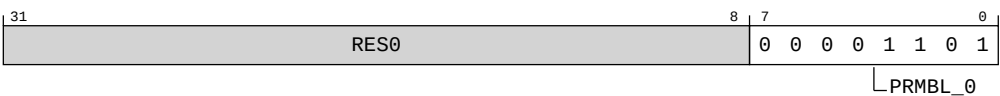


Table B-90: PMCIDR0 bit descriptions

| Bits   | Name    | Description                    | Reset |
|--------|---------|--------------------------------|-------|
| [31:8] | RES0    | Reserved                       | RES0  |
| [7:0]  | PRMBL_0 | Preamble.<br><b>0b00001101</b> | 0x0D  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFF0  | PMCIDR0  | None  |

This interface is accessible as follows:

**When !IsCorePowered()**

ERROR

**Otherwise**

RO

B.3.23 PMCIDR1, Performance Monitors Component Identification Register 1

Provides information to identify a Performance Monitor component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

Attributes

Width

32

Component

PMU

Register offset

0xFF4

Access type

See bit descriptions

Reset value

```
xxxx xxxx xxxx xxxx xxxx xxxx 1001 0000
```



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-45: pmu\_pmcidr1 bit assignments



Table B-92: PMCIDR1 bit descriptions

| Bits   | Name    | Description                                               | Reset  |
|--------|---------|-----------------------------------------------------------|--------|
| [31:8] | RES0    | Reserved                                                  | RES0   |
| [7:4]  | CLASS   | Component class.<br><b>0b1001</b><br>CoreSight component. | 0b1001 |
| [3:0]  | PRMBL_1 | Preamble. <b>RAZ.</b><br><b>0b0000</b>                    | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFF4  | PMCIDR1  | None  |

This interface is accessible as follows:

When **!IsCorePowered()**

ERROR

Otherwise

RO

B.3.24 PMCIDR2, Performance Monitors Component Identification Register 2

Provides information to identify a Performance Monitor component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

Attributes

Width

32

Component

PMU

Register offset

0xFF8

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-46: pmu\_pmcidr2 bit assignments



Table B-94: PMCIDR2 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_2 | Preamble.<br>0b00000101 | 0x05  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFF8  | PMCIDR2  | None  |

This interface is accessible as follows:

**When !IsCorePowered()**

ERROR

**Otherwise**

RO

### B.3.25 PMCIDR3, Performance Monitors Component Identification Register 3

Provides information to identify a Performance Monitor component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

If FEAT\_DoPD is implemented, this register is in the Core power domain. If FEAT\_DoPD is not implemented, this register is in the Debug power domain.

This register is required for CoreSight compliance.

#### Attributes

**Width**

32

**Component**

PMU

**Register offset**

0xFFC

**Access type**

See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1011 | 0001 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-47: pmu\_pmcidr3 bit assignments

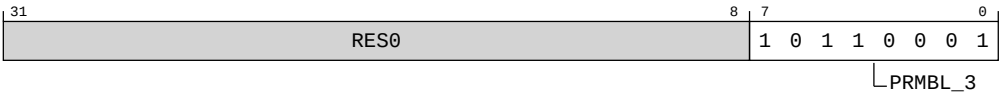


Table B-96: PMCIDR3 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_3 | Preamble.<br>0b10110001 | 0xB1  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| PMU       | 0xFFC  | PMCIDR3  | None  |

This interface is accessible as follows:

When !IsCorePowered()

ERROR

Otherwise

RO

B.4 External Debug registers summary

The following summary table provides an overview of all memory-mapped Debug registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

Table B-98: Debug registers summary

| Offset | Name  | Reset                      | Width  | Description                                |
|--------|-------|----------------------------|--------|--------------------------------------------|
| 0x020  | EDESR | See individual bit resets. | 32-bit | External Debug Event Status Register       |
| 0x024  | EDECR | See individual bit resets. | 32-bit | External Debug Execution Control Register  |
| 0x030  | EDWAR | See individual bit resets. | 64-bit | External Debug Watchpoint Address Register |

| Offset | Name               | Reset                      | Width  | Description                                         |
|--------|--------------------|----------------------------|--------|-----------------------------------------------------|
| 0x080  | DBGDTRRX_ELO       | See individual bit resets. | 32-bit | Debug Data Transfer Register, Receive               |
| 0x084  | EDITR              | See individual bit resets. | 32-bit | External Debug Instruction Transfer Register        |
| 0x088  | EDSCR              | See individual bit resets. | 32-bit | External Debug Status and Control Register          |
| 0x08C  | DBGDTRTX_ELO       | See individual bit resets. | 32-bit | Debug Data Transfer Register, Transmit              |
| 0x090  | EDRCR              | See individual bit resets. | 32-bit | External Debug Reserve Control Register             |
| 0x098  | EDECCR             | See individual bit resets. | 32-bit | External Debug Exception Catch Control Register     |
| 0x300  | OSLAR_EL1          | See individual bit resets. | 32-bit | OS Lock Access Register                             |
| 0x310  | EDPRCR             | See individual bit resets. | 32-bit | External Debug Power/Reset Control Register         |
| 0x314  | EDPRSR             | See individual bit resets. | 32-bit | External Debug Processor Status Register            |
| 0x400  | DBGBVR0_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                    |
| 0x408  | DBGBCR0_EL1        | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers                  |
| 0x410  | DBGBVR1_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                    |
| 0x418  | DBGBCR1_EL1        | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers                  |
| 0x420  | DBGBVR2_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                    |
| 0x428  | DBGBCR2_EL1        | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers                  |
| 0x430  | DBGBVR3_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                    |
| 0x438  | DBGBCR3_EL1        | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers                  |
| 0x440  | DBGBVR4_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                    |
| 0x448  | DBGBCR4_EL1        | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers                  |
| 0x450  | DBGBVR5_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Breakpoint Value Registers                    |
| 0x458  | DBGBCR5_EL1        | See individual bit resets. | 64-bit | Debug Breakpoint Control Registers                  |
| 0x800  | DBGWVR0_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                    |
| 0x808  | DBGWCR0_EL1        | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0x810  | DBGWVR1_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                    |
| 0x818  | DBGWCR1_EL1        | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0x820  | DBGWVR2_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                    |
| 0x828  | DBGWCR2_EL1        | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0x830  | DBGWVR3_EL1 [63:0] | See individual bit resets. | 64-bit | Debug Watchpoint Value Registers                    |
| 0x838  | DBGWCR3_EL1        | See individual bit resets. | 64-bit | Debug Watchpoint Control Registers                  |
| 0xD00  | MIDR_EL1           | See individual bit resets. | 32-bit | Main ID Register                                    |
| 0xD20  | EDPFR              | See individual bit resets. | 64-bit | External Debug Processor Feature Register           |
| 0xD28  | EDDFR              | See individual bit resets. | 64-bit | External Debug Feature Register                     |
| 0xD48  | EDDFR1             | See individual bit resets. | 64-bit | External Debug Feature Register 1                   |
| 0xD60  | EDAA32PFR          | See individual bit resets. | 64-bit | External Debug Auxiliary Processor Feature Register |
| 0xF00  | EDITCTRL           | See individual bit resets. | 32-bit | External Debug Integration mode Control register    |
| 0xFA0  | DBGCLAIMSET_EL1    | See individual bit resets. | 32-bit | Debug CLAIM Tag Set Register                        |
| 0xFA4  | DBGCLAIMCLR_EL1    | See individual bit resets. | 32-bit | Debug CLAIM Tag Clear Register                      |
| 0xFA8  | EDDEVAFF0          | See individual bit resets. | 32-bit | External Debug Device Affinity register 0           |
| 0xFAC  | EDDEVAFF1          | See individual bit resets. | 32-bit | External Debug Device Affinity register 1           |
| 0xFB0  | EDLAR              | See individual bit resets. | 32-bit | External Debug Lock Access Register                 |

| Offset | Name              | Reset                      | Width  | Description                                         |
|--------|-------------------|----------------------------|--------|-----------------------------------------------------|
| 0xFB4  | EDLSR             | See individual bit resets. | 32-bit | External Debug Lock Status Register                 |
| 0xFB8  | DBGAUTHSTATUS_EL1 | See individual bit resets. | 32-bit | Debug Authentication Status Register                |
| 0xFBC  | EDDEVARCH         | See individual bit resets. | 32-bit | External Debug Device Architecture Register         |
| 0xFC0  | EDDEVID2          | See individual bit resets. | 32-bit | External Debug Device ID register 2                 |
| 0xFC4  | EDDEVID1          | See individual bit resets. | 32-bit | External Debug Device ID Register 1                 |
| 0xFC8  | EDDEVID           | See individual bit resets. | 32-bit | External Debug Device ID register 0                 |
| 0xFCC  | EDDEVTYPE         | See individual bit resets. | 32-bit | External Debug Device Type register                 |
| 0xFD0  | EDPIDR4           | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 4 |
| 0xFE0  | EDPIDR0           | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 0 |
| 0xFE4  | EDPIDR1           | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 1 |
| 0xFE8  | EDPIDR2           | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 2 |
| 0xFEC  | EDPIDR3           | See individual bit resets. | 32-bit | External Debug Peripheral Identification Register 3 |
| 0xFF0  | EDCIDR0           | See individual bit resets. | 32-bit | External Debug Component Identification Register 0  |
| 0xFF4  | EDCIDR1           | See individual bit resets. | 32-bit | External Debug Component Identification Register 1  |
| 0xFF8  | EDCIDR2           | See individual bit resets. | 32-bit | External Debug Component Identification Register 2  |
| 0xFFC  | EDCIDR3           | See individual bit resets. | 32-bit | External Debug Component Identification Register 3  |

### B.4.1 EDRCR, External Debug Reserve Control Register

This register is used to allow imprecise entry to Debug state and clear sticky bits in ext-EDSCR.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

Debug

##### Register offset

0x090

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |





Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-48: ext\_edrcr bit assignments

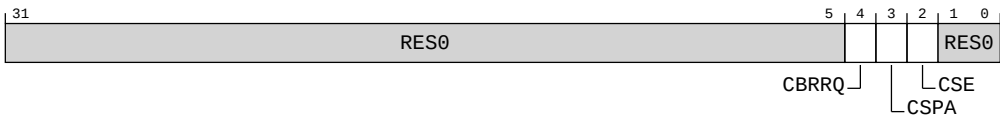


Table B-99: EDRCR bit descriptions

| Bits   | Name  | Description                                                                                                                                                                                                                 | Reset |
|--------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:5] | RES0  | Reserved                                                                                                                                                                                                                    | RES0  |
| [4]    | CBRRQ | This feature is not supported. Writes to this bit are ignored<br><br>0b0<br>No action.                                                                                                                                      | x     |
| [3]    | CSPA  | Clear Sticky Pipeline Advance. This bit is used to clear the ext-EDSCR.PipeAdv bit to 0.<br><br>0b0<br>No action.<br><br>0b1<br>Clear the ext-EDSCR.PipeAdv bit to 0.                                                       | x     |
| [2]    | CSE   | Clear Sticky Error. Used to clear the ext-EDSCR cumulative error bits to 0.<br><br>0b0<br>No action.<br><br>0b1<br>Clear the ext-EDSCR.{TXU, RXO, ERR} bits, and, if the PE is in Debug state, the ext-EDSCR.ITO bit, to 0. | x     |
| [1:0]  | RES0  | Reserved                                                                                                                                                                                                                    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0x090  | EDRCR    | None  |

This interface is accessible as follows:

When IsCorePowered() && !DoubleLockStatus() && !OSLockStatus() && SoftwareLockStatus()  
    WI

When IsCorePowered() && !DoubleLockStatus() && !OSLockStatus() && !SoftwareLockStatus()  
    WO

Otherwise  
    ERROR

### B.4.2 EDPRCR, External Debug Power/Reset Control Register

Controls the PE functionality related to powerup, reset, and powerdown.

#### Configurations

CORENPDRQ is the only field that is mapped between the EDPRCR and DBGPRCR and DBGPRCR\_EL1.

#### Attributes

Width

32

Component

Debug

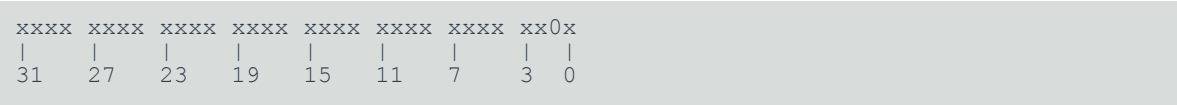
Register offset

0x310

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-49: ext\_edprcr bit assignments

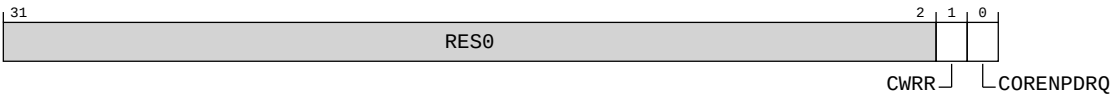


Table B-101: EDPRCR bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:2] | RES0 | Reserved    | RES0  |

| Bits | Name      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset          |
|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| [1]  | CWRR      | <p>Warm reset request.</p> <p>The extent of the reset is <b>IMPLEMENTATION DEFINED</b>, but must be one of:</p> <ul style="list-style-type: none"> <li>The request is ignored.</li> <li>Only this PE is Warm reset.</li> <li>This PE and other components of the system, possibly including other PEs, are Warm reset.</li> </ul> <p>Arm deprecates use of this bit, and recommends that implementations ignore the request.</p> <p><b>0b0</b><br/>No action.</p> <p><b>0b1</b><br/>Request Warm reset.</p> <p><b>When OSLockStatus()    SoftwareLockStatus()</b><br/>Access to this field is: <b>RAZ/WI</b></p> <p><b>Otherwise</b><br/>Access to this field is: <b>WO/RAZ</b></p> | 0b0            |
| [0]  | CORENPDRQ | <p>This field is in the Core power domain, and permitted accesses to this field map to the AArch64-DBGPRCR_EL1.CORENPDRQ field.</p> <p><b>0b0</b><br/>If the system responds to a powerdown request, it powers down Core power domain.</p> <p><b>0b1</b><br/>If the system responds to a powerdown request, it does not powerdown the Core power domain, but instead emulates a powerdown of that domain.</p> <p><b>When OSLockStatus()</b><br/>Access to this field is: <b>UNKNOWN/WI</b></p> <p><b>When SoftwareLockStatus()</b><br/>Access to this field is: <b>RO</b></p> <p><b>Otherwise</b><br/>Access to this field is: <b>RW</b></p>                                        | x <sup>3</sup> |

## Accessibility

On permitted accesses to the register, other access controls affect the behavior of some fields. See the field descriptions for more information.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0x310  | EDPRCR   | None  |

This interface is accessible as follows:

**When IsCorePowered() && SoftwareLockStatus()**

RO

<sup>3</sup> On a Cold reset, if the powerup request is implemented and the powerup request has been asserted, this field is an **IMPLEMENTATION DEFINED** choice of 0 or 1. If the powerup request is not asserted, this field is set to 0.

When IsCorePowered() && !SoftwareLockStatus()  
RW  
Otherwise  
ERROR

B.4.3 MIDR\_EL1, Main ID Register

Provides identification information for the PE, including an implementer code for the device and a device ID number.

Configurations

This register is available in all configurations.

Attributes

Width  
32  
Component  
Debug  
Register offset  
0xD00  
Access type  
See bit descriptions  
Reset value  
0100 0001 0000 1111 1101 1000 0101 0010

Bit descriptions

Figure B-50: ext\_midr\_el1 bit assignments

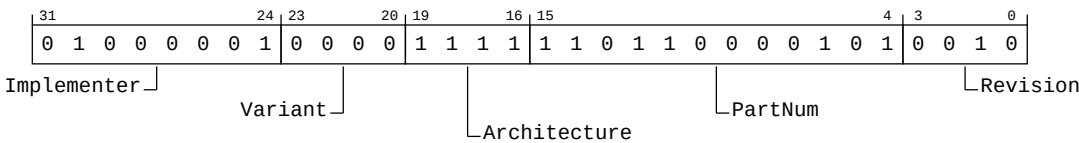


Table B-103: MIDR\_EL1 bit descriptions

| Bits    | Name        | Description                                                                      | Reset |
|---------|-------------|----------------------------------------------------------------------------------|-------|
| [31:24] | Implementer | Indicates the implementer code. This value is:<br><br>0b01000001<br>Arm Limited. | 0x41  |

| Bits    | Name         | Description                                                                                                                                                                                                                                    | Reset  |
|---------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [23:20] | Variant      | Variant number. Typically, this field is used to distinguish between different product variants, or major revisions of a product.<br><br><b>0b0000</b><br>rOp2                                                                                 | 0b0000 |
| [19:16] | Architecture | Indicates the architecture code. This value is:<br><br><b>0b1111</b><br>Architecture is defined by ID registers                                                                                                                                | 0b1111 |
| [15:4]  | PartNum      | Primary Part Number for the device.<br><br>On processors implemented by Arm, if the top four bits of the primary part number are 0x0 or 0x7, the variant and architecture are encoded differently.<br><br><b>0b110110000101</b><br>Cortex-X925 | 0xD85  |
| [3:0]   | Revision     | Revision number for the device.<br><br><b>0b0010</b><br>rOp2                                                                                                                                                                                   | 0b0010 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xD00  | MIDR_EL1 | None  |

This interface is accessible as follows:

**When IsCorePowered() && !DoubleLockStatus()**  
RO

**Otherwise**  
ImplementationDefined

B.4.4 EDPFR, External Debug Processor Feature Register

Provides information about implemented PE features.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

**Width**  
64

**Component**  
Debug

Register offset

0xD20

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | 0001 | xxxx | 0001 | 0001 | xxxx | xxxx | 0001 | 0001 | 0001 | 0001 | 0001 | 0001 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-51: ext\_edpfr bit assignments

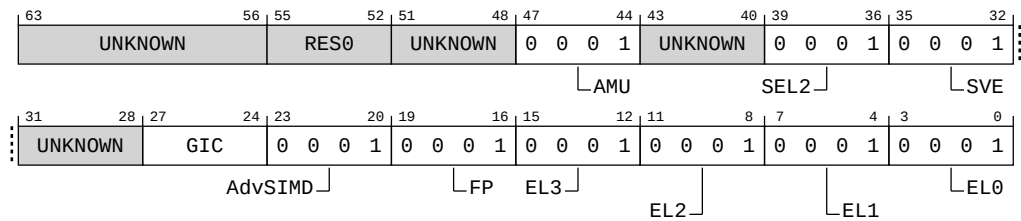


Table B-105: EDPFR bit descriptions

| Bits    | Name    | Description                                                                                 | Reset   |
|---------|---------|---------------------------------------------------------------------------------------------|---------|
| [63:56] | UNKNOWN | Reserved                                                                                    | UNKNOWN |
| [55:52] | RES0    | Reserved                                                                                    | RES0    |
| [51:48] | UNKNOWN | Reserved                                                                                    | UNKNOWN |
| [47:44] | AMU     | Activity Monitors Extension. This value is :<br><b>0b0001</b><br>FEAT_AMUv1 is implemented. | 0b0001  |
| [43:40] | UNKNOWN | Reserved                                                                                    | UNKNOWN |
| [39:36] | SEL2    | Secure EL2. This value is :<br><b>0b0001</b><br>Secure EL2 is implemented.                  | 0b0001  |
| [35:32] | SVE     | Scalable Vector Extension. This value is :<br><b>0b0001</b><br>SVE is implemented.          | 0b0001  |
| [31:28] | UNKNOWN | Reserved                                                                                    | UNKNOWN |

| Bits    | Name    | Description                                                                                                                                                                                                                                        | Reset                                                                           |
|---------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [27:24] | GIC     | System register GIC interface support. Defined values are:<br><br><b>0b0000</b><br>When Port GICCDISABLE is High, GIC CPU interface is disabled.<br><br><b>0b0011</b><br>When Port GICCDISABLE is Low, GIC (version 4.1) CPU interface is enabled. | The reset values can be the following: 0b0000, 0b0011, respective to the value. |
| [23:20] | AdvSIMD | Advanced SIMD. This value is:<br><br><b>0b0001</b><br>As for 0b0000, and also includes support for half-precision floating-point arithmetic.                                                                                                       | 0b0001                                                                          |
| [19:16] | FP      | Floating Point. This value is:<br><br><b>0b0001</b><br>As for 0b0000, and also includes support for half-precision floating-point arithmetic.                                                                                                      | 0b0001                                                                          |
| [15:12] | EL3     | AArch64 EL3 Exception level handling<br><br><b>0b0001</b><br>EL3 can be executed in AArch64 state only.                                                                                                                                            | 0b0001                                                                          |
| [11:8]  | EL2     | AArch64 EL2 Exception level handling<br><br><b>0b0001</b><br>EL2 can be executed in AArch64 state only.                                                                                                                                            | 0b0001                                                                          |
| [7:4]   | EL1     | AArch64 EL1 Exception level handling<br><br><b>0b0001</b><br>EL1 can be executed in AArch64 state only.                                                                                                                                            | 0b0001                                                                          |
| [3:0]   | ELO     | AArch64 ELO Exception level handling<br><br><b>0b0001</b><br>ELO can be executed in AArch64 state only.                                                                                                                                            | 0b0001                                                                          |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xD20  | EDPFR    | None  |

This interface is accessible as follows:

**When IsCorePowered() && !DoubleLockStatus()**

RO

**When !IsCorePowered()**

ERROR

**Otherwise**

ImplementationDefined

### B.4.5 EDDFR, External Debug Feature Register

Provides top level information about the debug system.



Debuggers must use ext-EDDEVARCH to determine the Debug architecture version.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is available in all configurations.

#### Attributes

**Width**

64

**Component**

Debug

**Register offset**

0xD28

**Access type**

See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | 0000 | xxxx | xxxx | xxxx | 0001 | xxxx | xxxx | 0001 | 0000 | 0011 | 0000 | 0101 | 0111 | 0001 | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.



## Bit descriptions

Figure B-52: ext\_eddfr bit assignments

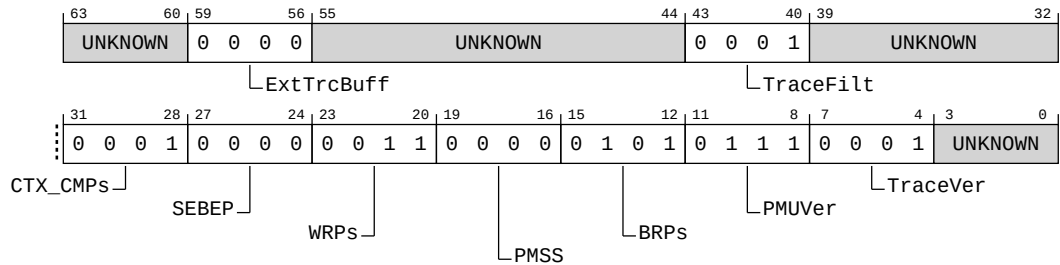


Table B-107: EDDFR bit descriptions

| Bits    | Name       | Description                                                                                                                                                                  | Reset   |
|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [63:60] | UNKNOWN    | Reserved                                                                                                                                                                     | UNKNOWN |
| [59:56] | ExtTrcBuff | Trace Buffer External Mode Extension. Defined values are:<br><br><b>0b0000</b><br>Trace Buffer Extension not implemented or Trace Buffer External Mode not implemented.      | 0b0000  |
| [55:44] | UNKNOWN    | Reserved                                                                                                                                                                     | UNKNOWN |
| [43:40] | TraceFilt  | Armv8.4 Self-hosted Trace Extension version. This value is :<br><br><b>0b0001</b><br>Armv8.4 Self-hosted Trace Extension is implemented.                                     | 0b0001  |
| [39:32] | UNKNOWN    | Reserved                                                                                                                                                                     | UNKNOWN |
| [31:28] | CTX_CMPS   | Number of breakpoints that are context-aware, minus 1.<br><br><b>0b0001</b><br>Two context-aware breakpoints are included                                                    | 0b0001  |
| [27:24] | SEBEP      | This field either has the same value as AArch64-ID_AA64DFR0_EL1.SEBEP or reads as zero.<br><br><b>0b0000</b><br>Synchronous-exception-based event profiling not implemented. | 0b0000  |
| [23:20] | WRPs       | Number of watchpoints, minus 1.<br><br><b>0b0011</b><br>Four Watchpoints                                                                                                     | 0b0011  |
| [19:16] | PMSS       | This field either has the same value as AArch64-ID_AA64DFR0_EL1.PMSS or reads as zero.<br><br><b>0b0000</b><br>PMU snapshot extension not implemented.                       | 0b0000  |
| [15:12] | BRPs       | Number of breakpoints, minus 1.<br><br><b>0b0101</b><br>Six Breakpoints                                                                                                      | 0b0101  |

| Bits   | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset   |
|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [11:8] | PMUVer   | <p>Performance Monitors Extension version.</p> <p>This field does not follow the standard ID scheme, but uses the alternative ID scheme described in <i>Alternative ID scheme used for the Performance Monitors Extension version</i> in the <a href="#">Arm® Architecture Reference Manual for A-profile architecture</a></p> <p>Defined values are:</p> <p><b>0b0111</b></p> <p>PMUv3 for Armv8.7. As 0b0110, and adds support for:</p> <ul style="list-style-type: none"> <li>The PMU.PMCR_ELO.FZO and, if EL2 is implemented, AArch64-MDCR_EL2.HPMFZO controls.</li> <li>If EL3 is implemented, the AArch64-MDCR_EL3.{MPMX,MCCD} controls.</li> </ul> | 0b0111  |
| [7:4]  | TraceVer | <p>Trace support. Indicates whether System register interface to a PE trace unit is implemented.</p> <p><b>0b0001</b></p> <p>Trace unit System registers implemented.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0b0001  |
| [3:0]  | UNKNOWN  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | UNKNOWN |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xD28  | EDDFR    | None  |

This interface is accessible as follows:

**When IsCorePowered() && !DoubleLockStatus()**

RO

**When !IsCorePowered()**

ERROR

**Otherwise**

ImplementationDefined

## B.4.6 EDDFR1, External Debug Feature Register 1

Provides top level information about the debug system in AArch64.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Component

Debug

**Register offset**

0xD48

**Access type**

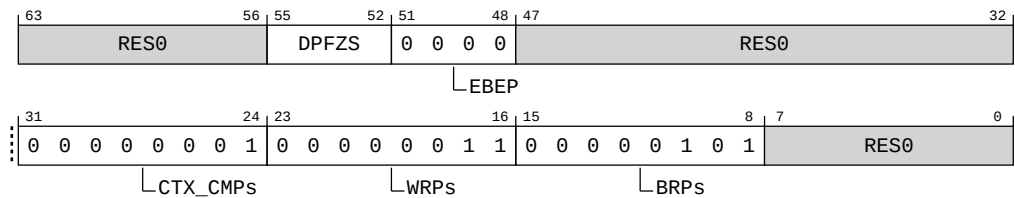
See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | 0000 | xxxx | xxxx | xxxx | xxxx | 0000 | 0001 | 0000 | 0011 | 0000 | 0101 | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

**Bit descriptions****Figure B-53: ext\_eddfr1 bit assignments****Table B-109: EDDFR1 bit descriptions**

| Bits    | Name     | Description                                                                                                                                                 | Reset  |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:56] | RES0     | Reserved                                                                                                                                                    | RES0   |
| [55:52] | DPFZS    | This field either has the same value as AArch64-ID_AA64DFR1_EL1.DPFZS or reads as zero.                                                                     | xxxx   |
| [51:48] | EBEP     | This field either has the same value as AArch64-ID_AA64DFR1_EL1.EBEP or reads as zero.<br><b>0b0000</b><br>Exception-based event profiling not implemented. | 0b0000 |
| [47:32] | RES0     | Reserved                                                                                                                                                    | RES0   |
| [31:24] | CTX_CMPS | Context-aware breakpoints. Defined values are:<br><b>0b00000001</b><br>Number of breakpoints that are context-aware minus 1.                                | 0x01   |
| [23:16] | WRPs     | Watchpoints. Defined values are:<br><b>0b00000011</b><br>Number of watchpoints minus 1.                                                                     | 0x03   |
| [15:8]  | BRPs     | Breakpoints. Defined values are:<br><b>0b00000101</b><br>Number of breakpoints minus 1.                                                                     | 0x05   |
| [7:0]   | RES0     | Reserved                                                                                                                                                    | RES0   |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xD48  | EDDFR1   | None  |

This interface is accessible as follows:

**When IsCorePowered() && !DoubleLockStatus() && (!IsZero(EDDFR1) || !OSLockStatus())**  
RO

**When !IsCorePowered()**  
ERROR

**Otherwise**  
ImplementationDefined

B.4.7 EDDEVARCH, External Debug Device Architecture Register

Identifies the programmers' model architecture of the external debug component.

Configurations

This register is available in all configurations.

Attributes

**Width**  
32

**Component**  
Debug

**Register offset**  
0xFBC

**Access type**  
See bit descriptions

**Reset value**  
0100 0111 0111 0000 1001 1010 0001 0101

Bit descriptions

Figure B-54: ext\_eddevarch bit assignments

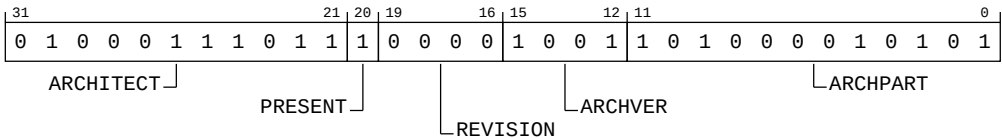


**Table B-111: EDDEVARCH bit descriptions**

| Bits    | Name      | Description                                                                                                                                                                                                                        | Reset         |
|---------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [31:21] | ARCHITECT | Defines the architecture of the component. For debug, this is Arm Limited.<br><br>Bits [31:28] are the JEP106 continuation code, 0x4.<br><br>Bits [27:21] are the JEP106 ID code, 0x3B.<br><br><b>0b01000111011</b>                | 0b01000111011 |
| [20]    | PRESENT   | Indicates that the DEVARCH is present.<br><br><b>0b1</b>                                                                                                                                                                           | 0b1           |
| [19:16] | REVISION  | Defines the architecture revision. For architectures defined by Arm this is the minor revision.<br><br>For debug, the revision defined by Armv8 is 0x0.<br><br>All other values are reserved.<br><br><b>0b0000</b>                 | 0b0000        |
| [15:12] | ARCHVER   | Defines the architecture version of the component. This is the same value as AArch64-ID_AA64DFRO_EL1.DebugVer and AArch32-DBGDIDR.Version. This value is :<br><br><b>0b1001</b><br><br>Armv8.4 debug architecture, FEAT_Debugv8p4. | 0b1001        |
| [11:0]  | ARCHPART  | The fields ARCHVER and ARCHPART together form the field ARCHID, so that ARCHPART is ARCHID[11:0].<br><br><b>0b101000010101</b><br><br>Armv8-A debug architecture.                                                                  | 0xA15         |

### Accessibility

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| Debug     | 0xFBC  | EDDEVARCH | None  |

This interface is accessible as follows:

**When IsCorePowered()**

RO

**Otherwise**

ERROR

## B.4.8 EDDEVID2, External Debug Device ID register 2

Reserved for future descriptions of features of the debug implementation.

### Configurations

This register is available in all configurations.

Attributes

Width

32

Component

Debug

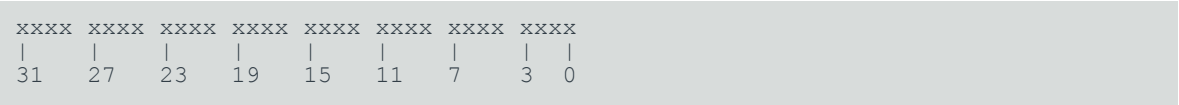
Register offset

0xFC0

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-55: ext\_eddevid2 bit assignments

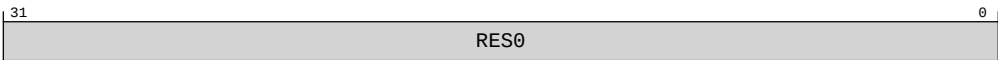


Table B-113: EDDEVID2 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFC0  | EDDEVID2 | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

B.4.9 EDDEVID1, External Debug Device ID Register 1

Provides extra information for external debuggers about features of the debug implementation.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

Debug

Register offset

0xFC4

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-56: ext\_eddevid1 bit assignments

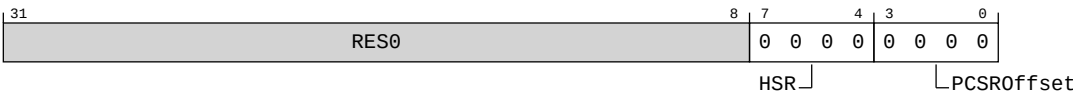


Table B-115: EDDEVID1 bit descriptions

| Bits   | Name | Description                                                                                                                                                                                                                                 | Reset  |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0 | Reserved                                                                                                                                                                                                                                    | RES0   |
| [7:4]  | HSR  | Indicates support for the External Debug Halt Status Register, ext-EDHSR. Defined values are:<br><br>0b0000<br><br>ext-EDHSR not implemented, and the PE follows behaviors consistent with all of the ext-EDHSR fields having a zero value. | 0b0000 |

| Bits  | Name       | Description                                                                                                                                                                  | Reset  |
|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [3:0] | PCSROffset | Indicates the offset applied to PC samples returned by reads of ext-EDPCSR. Permitted values of this field in Armv8 are:<br><br><b>0b0000</b><br>ext-EDPCSR not implemented. | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFC4  | EDDEVID1 | None  |

This interface is accessible as follows:

**When IsCorePowered()**

RO

**Otherwise**

ERROR

B.4.10 EDDEVID, External Debug Device ID register 0

Provides extra information for external debuggers about features of the debug implementation.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

Debug

Register offset

0xFC8

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | 0000 | xxxx | xxxx | xxxx | xxxx | 0001 | 0000 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.



Bit descriptions

Figure B-57: ext\_eddevid bit assignments

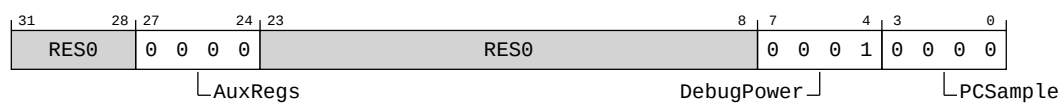


Table B-117: EDDEVID bit descriptions

| Bits    | Name       | Description                                                                                                                                                                                             | Reset  |
|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:28] | RES0       | Reserved                                                                                                                                                                                                | RES0   |
| [27:24] | AuxRegs    | Indicates support for Auxiliary registers.<br><br>0b0000<br>None supported.                                                                                                                             | 0b0000 |
| [23:8]  | RES0       | Reserved                                                                                                                                                                                                | RES0   |
| [7:4]   | DebugPower | Indicates support for the ARMv8.3-DoPD feature.<br><br>0b0001<br>FEAT_DoPD implemented. All registers in the external debug interface register map are implemented in the Core power domain.            | 0b0001 |
| [3:0]   | PCSample   | Indicates the level of PC Sample-based Profiling support using external debug registers.<br><br>0b0000<br>PC Sample-based Profiling Extension is not implemented in the external debug registers space. | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFC8  | EDDEVID  | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

B.4.11 EDDEVTYPE, External Debug Device Type register

Indicates to a debugger that this component is part of a PE's debug logic.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

Debug

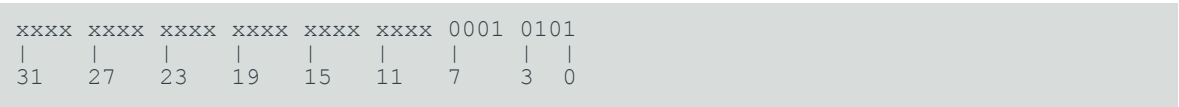
Register offset

0xFCC

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-58: ext\_eddevtype bit assignments



Table B-119: EDDEVTYPE bit descriptions

| Bits   | Name  | Description                                                      | Reset  |
|--------|-------|------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                         | RES0   |
| [7:4]  | SUB   | Subtype. Indicates this is a component within a PE.<br>0b0001    | 0b0001 |
| [3:0]  | MAJOR | Major type. Indicates this is a debug logic component.<br>0b0101 | 0b0101 |

Accessibility

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| Debug     | 0xFCC  | EDDEVTYPE | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

### B.4.12 EDPIDR4, External Debug Peripheral Identification Register 4

Provides information to identify an external debug component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is required for CoreSight compliance.

#### Attributes

**Width**

32

**Component**

Debug

**Register offset**

0xFD0

**Access type**

See bit descriptions

**Reset value**



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-59: ext\_edpidr4 bit assignments



Table B-121: EDPIDR4 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:8] | RES0 | Reserved    | RES0  |

| Bits  | Name  | Description                                                                                                                                                                               | Reset  |
|-------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [7:4] | SIZE  | Size of the component. Log <sub>2</sub> of the number of 4KB pages from the start of the component to the end of the component ID registers.<br><br>0b0000                                | 0b0000 |
| [3:0] | DES_2 | Designer, JEP106 continuation code, least significant nibble. For Arm Limited, this field is 0b0100.<br><br>0b0100<br><br>Arm Limited. This is bits[3:0] of the JEP106 continuation code. | 0b0100 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFD0  | EDPIDR4  | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

B.4.13 EDPIDR0, External Debug Peripheral Identification Register 0

Provides information to identify an external debug component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is required for CoreSight compliance.

Attributes

Width

32

Component

Debug

Register offset

0xFE0

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1000 | 0101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-60: ext\_edpidr0 bit assignments



Table B-123: EDPIDR0 bit descriptions

| Bits   | Name   | Description                                                                                               | Reset |
|--------|--------|-----------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0   | Reserved                                                                                                  | RES0  |
| [7:0]  | PART_0 | Part number, least significant byte.<br><br>0b10000101<br>Least Significant byte of the debug part number | 0x85  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFE0  | EDPIDR0  | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

B.4.14 EDPIDR1, External Debug Peripheral Identification Register 1

Provides information to identify an external debug component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is required for CoreSight compliance.

Attributes

Width

32

Component

Debug

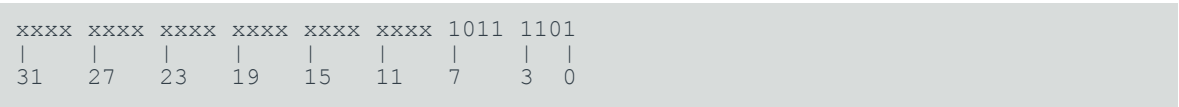
Register offset

0xFE4

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-61: ext\_edpidr1 bit assignments



Table B-125: EDPIDR1 bit descriptions

| Bits   | Name   | Description                                                                                                                                                                        | Reset  |
|--------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                                                                                                           | RES0   |
| [7:4]  | DES_0  | Designer, least significant nibble of JEP106 ID code. For Arm Limited, this field is 0b1011.<br><br>0b1011<br>Arm Limited. This is the least significant nibble of JEP106 ID code. | 0b1011 |
| [3:0]  | PART_1 | Part number, most significant nibble.<br><br>0b1101<br>Part number, most significant nibble.                                                                                       | 0b1101 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFE4  | EDPIDR1  | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise  
ERROR

B.4.15 EDPIDR2, External Debug Peripheral Identification Register 2

Provides information to identify an external debug component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is required for CoreSight compliance.

Attributes

Width  
32

Component  
Debug

Register offset  
0xFE8

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-62: ext\_edpidr2 bit assignments

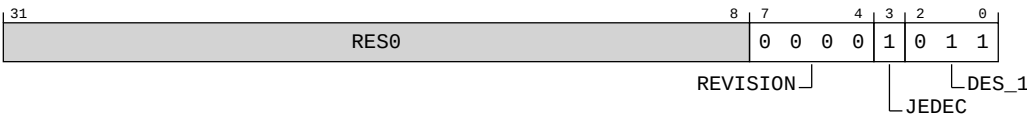


**Table B-127: EDPIDR2 bit descriptions**

| Bits   | Name     | Description                                                                                                                                                           | Reset  |
|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0     | Reserved                                                                                                                                                              | RES0   |
| [7:4]  | REVISION | Part major revision. Parts can also use this field to extend Part number to 16-bits.<br><br><b>0b0000</b><br>r0p2                                                     | 0b0000 |
| [3]    | JEDEC    | Indicates a JEP106 identity code is used.<br><br><b>0b1</b>                                                                                                           | 0b1    |
| [2:0]  | DES_1    | Designer, most significant bits of JEP106 ID code. For Arm Limited, this field is 0b011.<br><br><b>0b011</b><br>Arm Limited. This is bits[6:4] of the JEP106 ID code. | 0b011  |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFE8  | EDPIDR2  | None  |

This interface is accessible as follows:

#### When IsCorePowered()

RO

#### Otherwise

ERROR

## B.4.16 EDPIDR3, External Debug Peripheral Identification Register 3

Provides information to identify an external debug component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

This register is required for CoreSight compliance.

### Attributes

#### Width

32

#### Component

Debug

#### Register offset

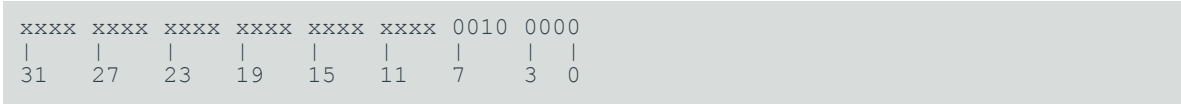
0xFEC

#### Access type

See bit descriptions



Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-63: ext\_edpidr3 bit assignments



Table B-129: EDPIDR3 bit descriptions

| Bits   | Name   | Description                                                                                                                                                          | Reset  |
|--------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                                                                                             | RES0   |
| [7:4]  | REVAND | Part minor revision. Parts using ext-EDPIDR2.REVISION as an extension to the Part number must use this field as a major revision number.<br><b>0b0010</b>            | 0b0010 |
| [3:0]  | CMOD   | Customer modified. Indicates someone other than the Designer has modified the component.<br><b>0b0000</b><br>The component is not modified from the original design. | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFEC  | EDPIDR3  | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

### B.4.17 EDCIDR0, External Debug Component Identification Register 0

Provides information to identify an external debug component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is required for CoreSight compliance.

#### Attributes

##### Width

32

##### Component

Debug

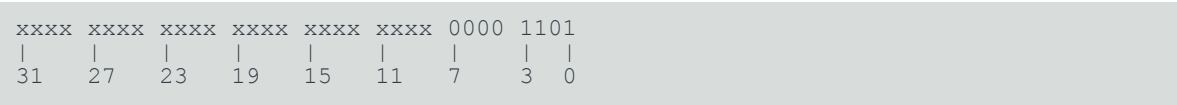
##### Register offset

0xFF0

##### Access type

See bit descriptions

##### Reset value



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-64: ext\_edcldr0 bit assignments

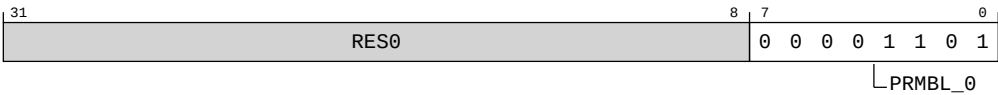


Table B-131: EDCIDR0 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_0 | Preamble.<br>0b00001101 | 0x0D  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFF0  | EDCIDR0  | None  |

This interface is accessible as follows:

**When IsCorePowered()**

RO

**Otherwise**

ERROR

B.4.18 EDCIDR1, External Debug Component Identification Register 1

Provides information to identify an external debug component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is required for CoreSight compliance.

Attributes

Width

32

Component

Debug

Register offset

0xFF4

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1001 | 0000 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-65: ext\_edcldr1 bit assignments

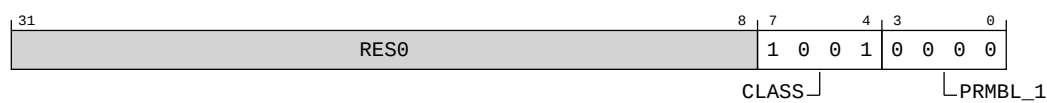


Table B-133: EDCIDR1 bit descriptions

| Bits   | Name    | Description                                               | Reset  |
|--------|---------|-----------------------------------------------------------|--------|
| [31:8] | RES0    | Reserved                                                  | RES0   |
| [7:4]  | CLASS   | Component class.<br><b>0b1001</b><br>CoreSight component. | 0b1001 |
| [3:0]  | PRMBL_1 | Preamble.<br><b>0b0000</b>                                | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFF4  | EDCIDR1  | None  |

This interface is accessible as follows:

**When IsCorePowered()**

RO

**Otherwise**

ERROR

B.4.19 EDCIDR2, External Debug Component Identification Register 2

Provides information to identify an external debug component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is required for CoreSight compliance.

Attributes

**Width**

32

**Component**

Debug

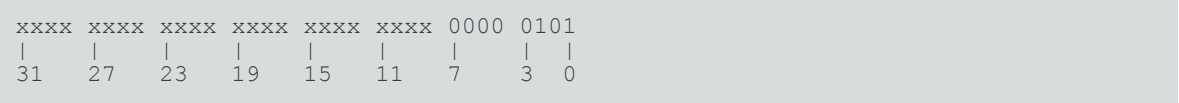
Register offset

0xFF8

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-66: ext\_edcldr2 bit assignments

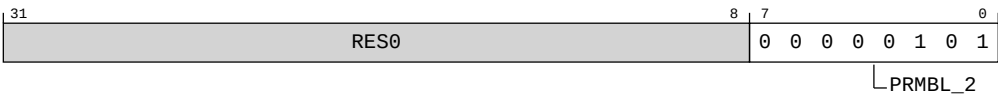


Table B-135: EDCIDR2 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_2 | Preamble.<br>0b00000101 | 0x05  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFF8  | EDCIDR2  | None  |

This interface is accessible as follows:

When IsCorePowered()

RO

Otherwise

ERROR

B.4.20 EDCIDR3, External Debug Component Identification Register 3

Provides information to identify an external debug component.

For more information, see *About the Component Identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is required for CoreSight compliance.

Attributes

Width

32

Component

Debug

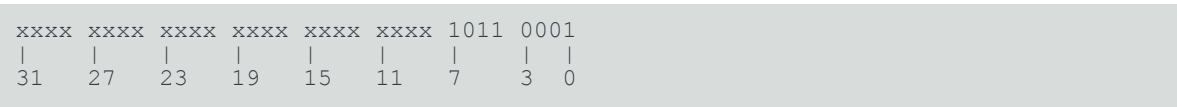
Register offset

0xFFC

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-67: ext\_edcldr3 bit assignments

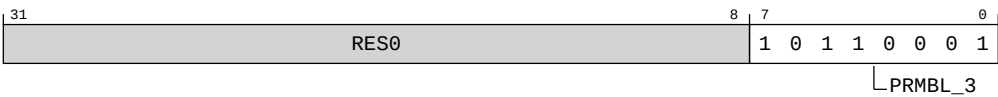


Table B-137: EDCIDR3 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_3 | Preamble.<br>0b10110001 | 0xB1  |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| Debug     | 0xFFC  | EDCIDR3  | None  |

This interface is accessible as follows:

### When IsCorePowered()

RO

### Otherwise

ERROR

## B.5 External CTI registers summary

The following summary table provides an overview of all memory-mapped CTI registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table B-139: CTI registers summary**

| Offset | Name                     | Reset                      | Width  | Description                              |
|--------|--------------------------|----------------------------|--------|------------------------------------------|
| 0x150  | CTIDEVCTL                | See individual bit resets. | 32-bit | CTI Device Control register              |
| 0xFE0  | <a href="#">CTIPIDR0</a> | See individual bit resets. | 32-bit | CTI Peripheral Identification Register 0 |
| 0xFE4  | <a href="#">CTIPIDR1</a> | See individual bit resets. | 32-bit | CTI Peripheral Identification Register 1 |

### B.5.1 CTIPIDR0, CTI Peripheral Identification Register 0

Provides information to identify a CTI component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations

This register is required for CoreSight compliance.

### Attributes

#### Width

32

Component

CTI

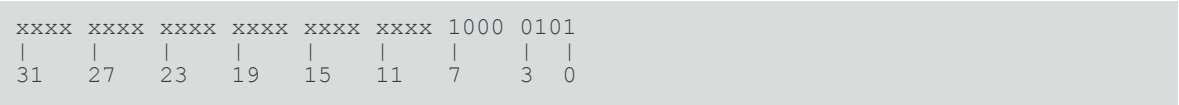
Register offset

0xFE0

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-68: ext\_ctipidr0 bit assignments



Table B-140: CTIPIDR0 bit descriptions

| Bits   | Name   | Description                                                                                                                   | Reset |
|--------|--------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0   | Reserved                                                                                                                      | RES0  |
| [7:0]  | PART_0 | Part number, least significant byte.<br><br><b>0b10000101</b><br>Cortex-X925 Core ROM table. Bits [7:0] of part number 0xD85. | 0x85  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| CTI       | 0xFE0  | CTIPIDR0 | None  |

This interface is accessible as follows:

RO



### B.5.2 CTIPIDR1, CTI Peripheral Identification Register 1

Provides information to identify a CTI component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is required for CoreSight compliance.

#### Attributes

**Width**

32

**Component**

CTI

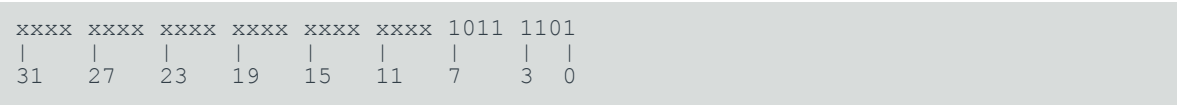
**Register offset**

0xFE4

**Access type**

RO

**Reset value**



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-69: ext\_ctipidr1 bit assignments

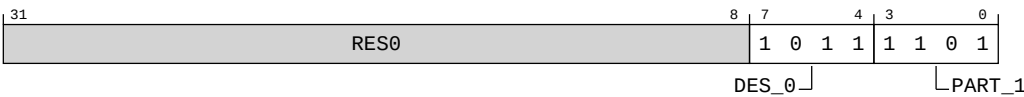


Table B-142: CTIPIDR1 bit descriptions

| Bits   | Name  | Description                                                                                                                                                         | Reset  |
|--------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                                                            | RES0   |
| [7:4]  | DES_0 | Designer, least significant nibble of JEP106 ID code. For Arm Limited, this field is 0b1011.<br><br>0b1011<br>Designer, least significant nibble of JEP106 ID code. | 0b1011 |

| Bits  | Name   | Description                                                                                                                 | Reset  |
|-------|--------|-----------------------------------------------------------------------------------------------------------------------------|--------|
| [3:0] | PART_1 | Part number, most significant nibble.<br><br><b>0b1101</b><br>Cortex-X925 Core ROM table. Bits [11:8] of part number 0xD85. | 0b1101 |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| CTI       | 0xFE4  | CTIPIDR1 | None  |

This interface is accessible as follows:

RO

## B.6 External AMU registers summary

The following summary table provides an overview of all memory-mapped AMU registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

If a register does not have a link in the summary table, you can find more information about this Architecturally defined register in the [Arm® Architecture Reference Manual for A-profile architecture](#).

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table B-144: AMU registers summary**

| Offset | Name                               | Reset                      | Width  | Description                                 |
|--------|------------------------------------|----------------------------|--------|---------------------------------------------|
| 0x0    | <a href="#">AMEVCNTR00 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x4    | <a href="#">AMEVCNTR00 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x8    | <a href="#">AMEVCNTR01 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0xC    | <a href="#">AMEVCNTR01 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x10   | <a href="#">AMEVCNTR02 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x14   | <a href="#">AMEVCNTR02 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x18   | <a href="#">AMEVCNTR03 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x1C   | <a href="#">AMEVCNTR03 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 0 |
| 0x100  | <a href="#">AMEVCNTR10 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1 |
| 0x104  | <a href="#">AMEVCNTR10 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1 |
| 0x108  | <a href="#">AMEVCNTR11 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1 |
| 0x10C  | <a href="#">AMEVCNTR11 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1 |
| 0x110  | <a href="#">AMEVCNTR12 [31:0]</a>  | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1 |
| 0x114  | <a href="#">AMEVCNTR12 [63:32]</a> | See individual bit resets. | 32-bit | Activity Monitors Event Counter Registers 1 |

| Offset | Name                        | Reset                      | Width  | Description                                              |
|--------|-----------------------------|----------------------------|--------|----------------------------------------------------------|
| 0x400  | <a href="#">AMEVTYPER00</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x404  | <a href="#">AMEVTYPER01</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x408  | <a href="#">AMEVTYPER02</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x40C  | <a href="#">AMEVTYPER03</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 0                 |
| 0x480  | <a href="#">AMEVTYPER10</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 1                 |
| 0x484  | <a href="#">AMEVTYPER11</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 1                 |
| 0x488  | <a href="#">AMEVTYPER12</a> | See individual bit resets. | 32-bit | Activity Monitors Event Type Registers 1                 |
| 0xC00  | AMCNTENSET0                 | See individual bit resets. | 32-bit | Activity Monitors Count Enable Set Register 0            |
| 0xC04  | AMCNTENSET1                 | See individual bit resets. | 32-bit | Activity Monitors Count Enable Set Register 1            |
| 0xC20  | AMCNTENCLR0                 | See individual bit resets. | 32-bit | Activity Monitors Count Enable Clear Register 0          |
| 0xC24  | AMCNTENCLR1                 | See individual bit resets. | 32-bit | Activity Monitors Count Enable Clear Register 1          |
| 0xCE0  | <a href="#">AMCGCR</a>      | See individual bit resets. | 32-bit | Activity Monitors Counter Group Configuration Register   |
| 0xE00  | <a href="#">AMCFGR</a>      | See individual bit resets. | 32-bit | Activity Monitors Configuration Register                 |
| 0xE04  | AMCR                        | See individual bit resets. | 32-bit | Activity Monitors Control Register                       |
| 0xE08  | <a href="#">AMIIDR</a>      | See individual bit resets. | 32-bit | Activity Monitors Implementation Identification Register |
| 0xFA8  | AMDEVAFF0                   | See individual bit resets. | 32-bit | Activity Monitors Device Affinity Register 0             |
| 0xFAC  | AMDEVAFF1                   | See individual bit resets. | 32-bit | Activity Monitors Device Affinity Register 1             |
| 0xFBC  | <a href="#">AMDEVARCH</a>   | See individual bit resets. | 32-bit | Activity Monitors Device Architecture Register           |
| 0xFCC  | <a href="#">AMDEVTYPE</a>   | See individual bit resets. | 32-bit | Activity Monitors Device Type Register                   |
| 0xFD0  | <a href="#">AMPIDR4</a>     | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 4   |
| 0xFE0  | <a href="#">AMPIDR0</a>     | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 0   |
| 0xFE4  | <a href="#">AMPIDR1</a>     | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 1   |
| 0xFE8  | <a href="#">AMPIDR2</a>     | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 2   |
| 0xFEC  | <a href="#">AMPIDR3</a>     | See individual bit resets. | 32-bit | Activity Monitors Peripheral Identification Register 3   |
| 0xFF0  | <a href="#">AMCIDR0</a>     | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 0    |
| 0xFF4  | <a href="#">AMCIDR1</a>     | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 1    |
| 0xFF8  | <a href="#">AMCIDR2</a>     | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 2    |
| 0xFFC  | <a href="#">AMCIDR3</a>     | See individual bit resets. | 32-bit | Activity Monitors Component Identification Register 3    |

## B.6.1 AMEVCNTR00, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counters.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

Component

AMU

Register offsets (2)

0x0,0x4

Access type

Read

R

Write

RESERVED

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure B-70: ext\_amevcntr00 bit assignments

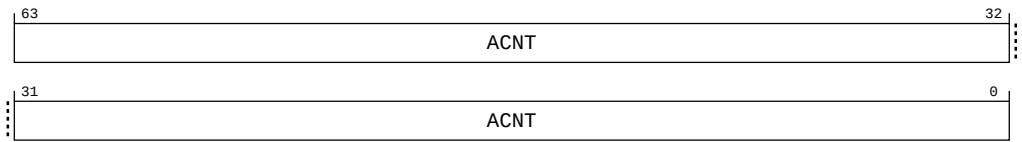


Table B-145: AMEVCNTR00 bit descriptions

| Bits   | Name | Description                                                                                                                                                                     | Reset              |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Architected activity monitor event counter n.<br><br>Value of architected activity monitor event counter n, where n is the number of this register and is a number from 0 to 3. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



Note

ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x0    | AMEVCNTR00 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x4    | AMEVCNTR00 | 63:32 |

This interface is accessible as follows:

RO

B.6.2 AMEVCNTR01, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

AMU

Register offsets (2)

0x8,0xC

Access type

Read

R

Write

RESERVED

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Bit descriptions

Figure B-71: ext\_amevcntr01 bit assignments

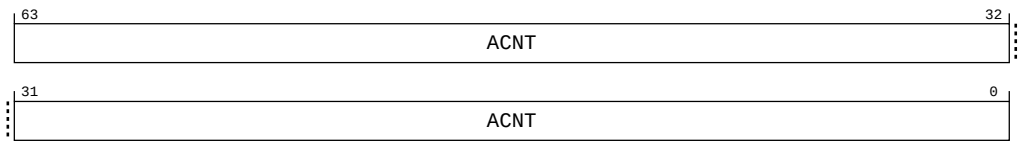


Table B-148: AMEVCNTR01 bit descriptions

| Bits   | Name | Description                                                                                                                                                                     | Reset              |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Architected activity monitor event counter n.<br><br>Value of architected activity monitor event counter n, where n is the number of this register and is a number from 0 to 3. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x8    | AMEVCNTR01 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0xC    | AMEVCNTR01 | 63:32 |

This interface is accessible as follows:

RO

B.6.3 AMEVCNTR02, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

AMU

Register offsets (2)

0x10,0x14

Access type

Read

R

Write

RESERVED

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure B-72: ext\_amevcntr02 bit assignments

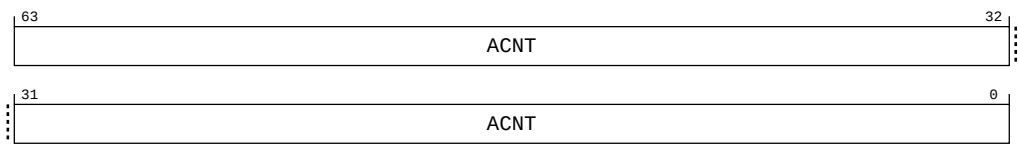


Table B-151: AMEVCNTR02 bit descriptions

| Bits   | Name | Description                                                                                                                                                                     | Reset              |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Architected activity monitor event counter n.<br><br>Value of architected activity monitor event counter n, where n is the number of this register and is a number from 0 to 3. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x10   | AMEVCNTR02 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x14   | AMEVCNTR02 | 63:32 |

This interface is accessible as follows:

RO



B.6.4 AMEVCNTR03, Activity Monitors Event Counter Registers 0

Provides access to the architected activity monitor event counters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

AMU

Register offsets (2)

0x18,0x1C

Access type

Read

R

Write

RESERVED

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Bit descriptions

Figure B-73: ext\_amevcntr03 bit assignments

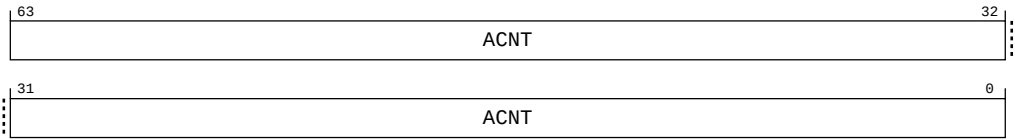


Table B-154: AMEVCNTR03 bit descriptions

| Bits   | Name | Description                                                                                                                                                                     | Reset              |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Architected activity monitor event counter n.<br><br>Value of architected activity monitor event counter n, where n is the number of this register and is a number from 0 to 3. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as **RES0**. See Access

requirements for reserved and unallocated registers in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVCNTR0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x18   | AMEVCNTR03 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x1C   | AMEVCNTR03 | 63:32 |

This interface is accessible as follows:

RO

B.6.5 AMEVCNTR10, Activity Monitors Event Counter Registers 1

Provides access to the auxiliary activity monitor event counters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

AMU

Register offsets (2)

0x100,0x104

Access type

Read

R

Write

RESERVED

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure B-74: ext\_amevcntr10 bit assignments

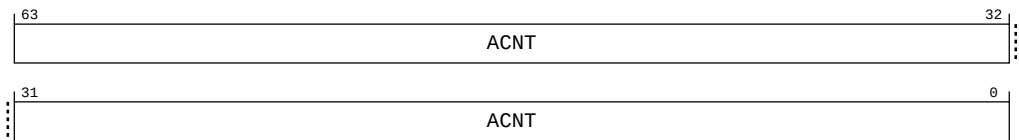


Table B-157: AMEVCNTR10 bit descriptions

| Bits   | Name | Description                                                                                                                                                                  | Reset              |
|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Auxiliary activity monitor event counter n.<br><br>Value of auxiliary activity monitor event counter n, where n is the number of this register and is a number from 0 to 15. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVCNTR1<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVCNTR1<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x100  | AMEVCNTR10 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x104  | AMEVCNTR10 | 63:32 |

This interface is accessible as follows:

RO

### B.6.6 AMEVCNTR11, Activity Monitors Event Counter Registers 1

Provides access to the auxiliary activity monitor event counters.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Component

AMU

##### Register offsets (2)

0x108,0x10C

##### Access type

###### Read

R

###### Write

RESERVED

##### Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure B-75: ext\_amevcntr11 bit assignments

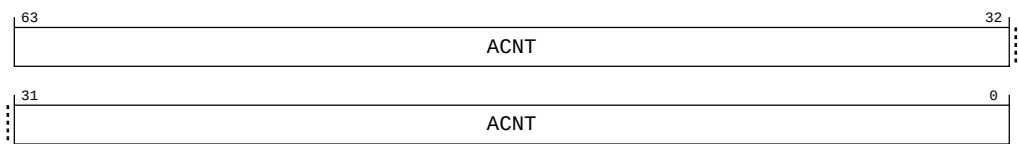


Table B-160: AMEVCNTR11 bit descriptions

| Bits   | Name | Description                                                                                                                                                                  | Reset              |
|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Auxiliary activity monitor event counter n.<br><br>Value of auxiliary activity monitor event counter n, where n is the number of this register and is a number from 0 to 15. | 0x0000000000000000 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVCNTR1<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVCNTR1<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x108  | AMEVCNTR11 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x10C  | AMEVCNTR11 | 63:32 |

This interface is accessible as follows:

RO

B.6.7 AMEVCNTR12, Activity Monitors Event Counter Registers 1

Provides access to the auxiliary activity monitor event counters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Component

AMU

Register offsets (2)

0x110,0x114

Access type

Read

R

Write

RESERVED

Reset value

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  
0000

Bit descriptions

Figure B-76: ext\_amevcntr12 bit assignments

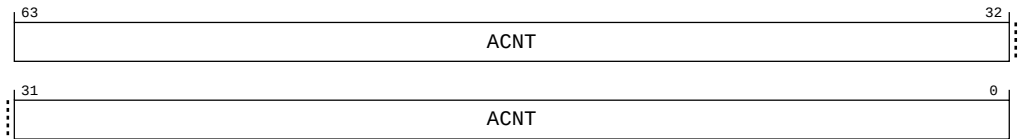


**Table B-163: AMEVCNTR12 bit descriptions**

| Bits   | Name | Description                                                                                                                                                                  | Reset              |
|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| [63:0] | ACNT | Auxiliary activity monitor event counter n.<br><br>Value of auxiliary activity monitor event counter n, where n is the number of this register and is a number from 0 to 15. | 0x0000000000000000 |

### Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVCNTR1<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

### Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVCNTR1<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x110  | AMEVCNTR12 | 31:0  |

This interface is accessible as follows:

RO

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| AMU       | 0x114  | AMEVCNTR12 | 63:32 |

This interface is accessible as follows:

RO

B.6.8 AMEVTYPER00, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter ext-AMEVCNTR0<n> counts.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0x400

Access type

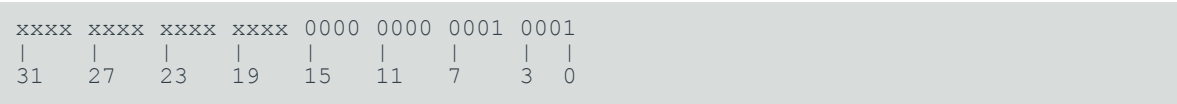
Read

R

Write

RESERVED

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-77: ext\_amevtyper00 bit assignments

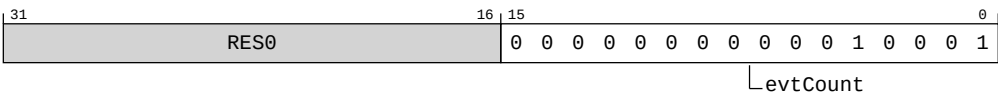


Table B-166: AMEVTYPER00 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [31:16] | RES0 | Reserved    | RES0  |



| Bits                | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset               |        |
|---------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|--------|
| [15:0]              | evtCount | <p>Event to count. The event number of the event that is counted by the architected activity monitor event counter ext-AMEVCNTR0&lt;n&gt;. The value of this field is architecturally mandated for each architected counter.</p> <p>The following table shows the mapping between required event numbers and the corresponding counters:</p> <table><tr><td>0b00000000000010001</td></tr></table> <p>Processor frequency cycles</p> | 0b00000000000010001 | 0x0011 |
| 0b00000000000010001 |          |                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |        |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x400  | AMEVTYPER00 | None  |

This interface is accessible as follows:

RO

B.6.9 AMEVTYPER01, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter ext-AMEVCNTR0<n> counts.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0x404

Access type

Read

R

Write

RESERVED

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-78: ext\_amevtyper01 bit assignments

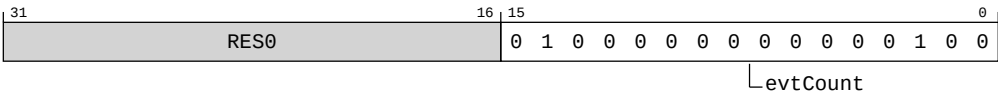


Table B-168: AMEVTYPER01 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Reset  |
|---------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0   |
| [15:0]  | evtCount | <div>Event to count. The event number of the event that is counted by the architected activity monitor event counter ext-AMEVCNTR0&lt;n&gt;. The value of this field is architecturally mandated for each architected counter.</div> <div>The following table shows the mapping between required event numbers and the corresponding counters:</div> <div>0b01000000000000100</div> <div>Constant frequency cycles</div> | 0x4004 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



Note

ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



Note

ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x404  | AMEVTYPER01 | None  |

This interface is accessible as follows:

RO

B.6.10 AMEVTYPER02, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter ext-AMEVCNTR0<n> counts.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0x408

Access type

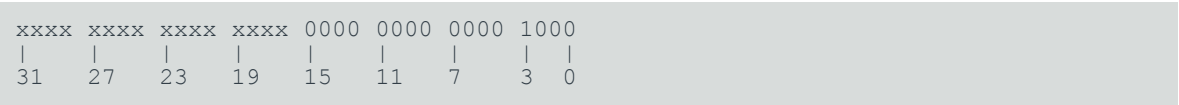
Read

R

Write

RESERVED

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-79: ext\_amevtyper02 bit assignments

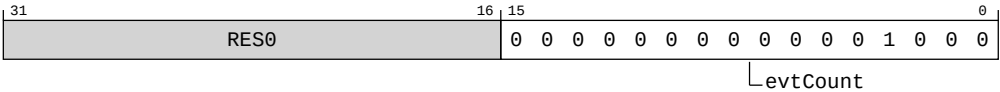


Table B-170: AMEVTYPER02 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                           | Reset  |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                              | RES0   |
| [15:0]  | evtCount | <div>The event number of the event that is counted by the architected activity monitor event counter ext-AMEVCNTR0&lt;n&gt;. The value of this field is architecturally mandated for each architected counter.</div> <div>The following table shows the mapping between required event numbers and the corresponding counters:</div> <div>0b0000000000000001000</div> <div>Instructions retired</div> | 0x0008 |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



Note

ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x408  | AMEVTYPER02 | None  |

This interface is accessible as follows:

RO

B.6.11 AMEVTYPER03, Activity Monitors Event Type Registers 0

Provides information on the events that an architected activity monitor event counter ext-AMEVCNTR0<n> counts.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0x40C

Access type

Read

R

Write

RESERVED

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | 0100 | 0000 | 0000 | 0101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-80: ext\_amevtyper03 bit assignments

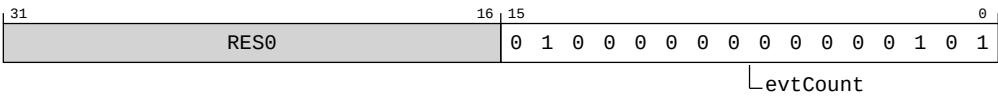


Table B-172: AMEVTYPER03 bit descriptions

| Bits               | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset              |        |
|--------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|--------|
| [31:16]            | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                        | RES0               |        |
| [15:0]             | evtCount | <div>Event to count. The event number of the event that is counted by the architected activity monitor event counter ext-AMEVCNTR0&lt;n&gt;. The value of this field is architecturally mandated for each architected counter.</div> <div>The following table shows the mapping between required event numbers and the corresponding counters:<br/><table><tr><td>0b0100000000000101</td></tr></table>Memory stall cycles</div> | 0b0100000000000101 | 0x4005 |
| 0b0100000000000101 |          |                                                                                                                                                                                                                                                                                                                                                                                                                                 |                    |        |

Access

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of architected activity monitor event counters, reads of AMEVTYPER0<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



ext-AMCGCR.CG0NC identifies the number of architected activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x40C  | AMEVTYPER03 | None  |

This interface is accessible as follows:

RO

### B.6.12 AMEVTYPER10, Activity Monitors Event Type Registers 1

Provides information on the events that an auxiliary activity monitor event counter ext-AMEVCNTR1<n> counts.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

AMU

##### Register offset

0x480

##### Access type

###### Read

R

###### Write

RESERVED

#### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | 0000 | 0011 | 0000 | 0000 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-81: ext\_amevtyper10 bit assignments

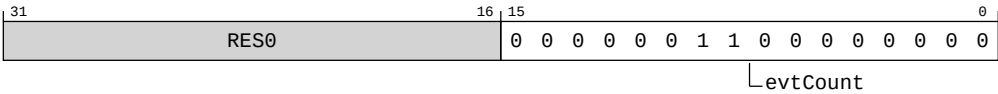


Table B-174: AMEVTYPER10 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                               | Reset  |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                  | RES0   |
| [15:0]  | evtCount | Event to count. The event number of the event that is counted by the auxiliary activity monitor event counter ext-AMEVCNTR1<n>.<br><br>It is <b>IMPLEMENTATION DEFINED</b> what values are supported by each counter.<br><br><b>0b00000001100000000</b><br>Gear 0 (MPMM bank 0) period threshold exceeded | 0x0300 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVTYPER1<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVTYPER1<n> are RAZ. Software must treat reserved accesses as **RES0**. See 'Access requirements for reserved and unallocated registers'.



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x480  | AMEVTYPER10 | None  |

This interface is accessible as follows:

RO



### B.6.13 AMEVTYPER11, Activity Monitors Event Type Registers 1

Provides information on the events that an auxiliary activity monitor event counter ext-AMEVCNTR1<n> counts.

## Configurations

This register is available in all configurations.

## Attributes

## Width

32

## Component

AMU

## Register offset

0x484

### Access type

## Read

R

## Write

RESERVED

## Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | 0000 | 0011 | 0000 | 0001 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

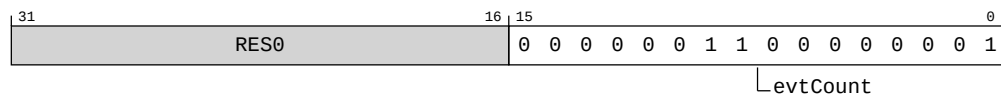


## Note

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure B-82: ext\_amevtyper11 bit assignments**



### Table B-176: AMEVTYPER11 bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [31:16] | RES0 | Reserved    | RES0  |

| Bits   | Name     | Description                                                                                                                                                                                                                                                                                               | Reset  |
|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:0] | evtCount | Event to count. The event number of the event that is counted by the auxiliary activity monitor event counter ext-AMEVCNTR1<n>.<br><br>It is <b>IMPLEMENTATION DEFINED</b> what values are supported by each counter.<br><br><b>0b00000001100000001</b><br>Gear 1 (MPMM bank 1) period threshold exceeded | 0x0301 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVTYPER1<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVTYPER1<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x484  | AMEVTYPER11 | None  |

This interface is accessible as follows:

RO

B.6.14 AMEVTYPER12, Activity Monitors Event Type Registers 1

Provides information on the events that an auxiliary activity monitor event counter ext-AMEVCNTR1<n> counts.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0x488

Access type

Read

R

Write

RESERVED

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-83: ext\_amevtyper12 bit assignments

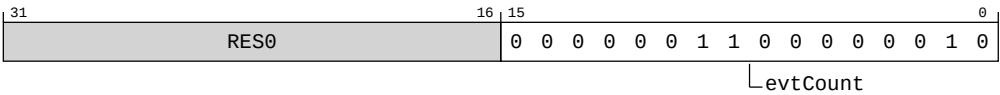


Table B-178: AMEVTYPER12 bit descriptions

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                         | Reset  |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:16] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                            | RES0   |
| [15:0]  | evtCount | <div>Event to count. The event number of the event that is counted by the auxiliary activity monitor event counter ext-AMEVCNTR1&lt;n&gt;.</div> <div>It is <b>IMPLEMENTATION DEFINED</b> what values are supported by each counter.</div> <div>0b00000001100000010</div> <div>Gear 2 (MPMM bank 2) period threshold exceeded</div> | 0x0302 |

Access

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVTYPER1<n> are RAZ. Software must treat reserved accesses as **RES0**. See *Access requirements for reserved and unallocated registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

Accessibility

If <n> is greater than or equal to the number of auxiliary activity monitor event counters, reads of AMEVTYPER1<n> are RAZ. Software must treat reserved accesses as RES0. See 'Access requirements for reserved and unallocated registers'.



Note

ext-AMCGCR.CG1NC identifies the number of auxiliary activity monitor event counters.

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| AMU       | 0x488  | AMEVTYPER12 | None  |

This interface is accessible as follows:

RO

B.6.15 AMCGCR, Activity Monitors Counter Group Configuration Register

Provides information on the number of activity monitor event counters implemented within each counter group.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

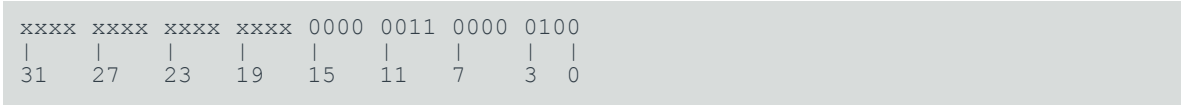
AMU

Register offset

0xCE0

Access type  
RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-84: ext\_amcgcr bit assignments

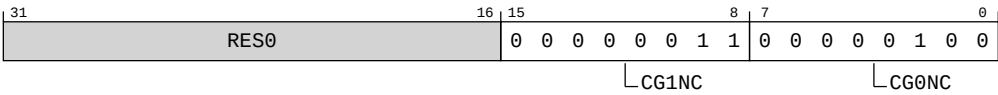


Table B-180: AMCGCR bit descriptions

| Bits    | Name  | Description                                                                                                                                                                                                                                                              | Reset |
|---------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:16] | RES0  | Reserved                                                                                                                                                                                                                                                                 | RES0  |
| [15:8]  | CG1NC | Counter Group 1 Number of Counters. The number of counters in the auxiliary counter group.<br><br>In an implementation that includes FEAT_AMUv1, the permitted range of values is 0 to 16.<br><br><b>0b00000011</b><br><br>Three counters in the auxiliary counter group | 0x03  |
| [7:0]   | CG0NC | Counter Group 0 Number of Counters. The number of counters in the architected counter group.<br><br><b>0b00000100</b>                                                                                                                                                    | 0x04  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xCE0  | AMCGCR   | None  |

This interface is accessible as follows:

RO

B.6.16 AMCFGR, Activity Monitors Configuration Register

Global configuration register for the activity monitors.

Provides information on supported features, the number of counter groups implemented, the total number of activity monitor event counters implemented, and the size of the counters. AMCFGR is applicable to both the architected and the auxiliary counter groups.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

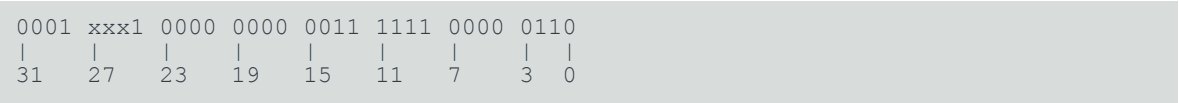
Register offset

0xE00

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-85: ext\_amcfgr bit assignments



Table B-182: AMCFGR bit descriptions

| Bits    | Name | Description                                                                                                                                             | Reset  |
|---------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:28] | NCG  | Defines the number of counter groups. The following value is specified for this product.<br><br><b>0b0001</b><br><br>Two counter groups are implemented | 0b0001 |

| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                            | Reset       |
|---------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [27:25] | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                                               | <b>RES0</b> |
| [24]    | HDBG        | Halt-on-debug supported.<br><br>This feature must be supported, and so this bit is 0b1.<br><br><b>0b1</b><br>ext-AMCR.HDBG is read/write.                                                                                                                                                                                                                                                              | 0b1         |
| [23:14] | <b>RAZ</b>  | Reserved                                                                                                                                                                                                                                                                                                                                                                                               | <b>RAZ</b>  |
| [13:8]  | SIZE        | Defines the size of activity monitor event counters.<br><br>The size of the activity monitor event counters implemented by the Activity Monitors Extension is [AMCFGR.SIZE + 1].<br><br>The counters are 64-bit.<br><br><b>Note:</b><br>Software also uses this field to determine the spacing of counters in the memory-map. The counters are at doubleword-aligned addresses.<br><br><b>0b111111</b> | 0b111111    |
| [7:0]   | N           | Defines the number of activity monitor event counters.<br><br>The total number of counters implemented in all groups by the Activity Monitors Extension is [AMCFGR.N + 1].<br><br><b>0b00000110</b><br>Seven activity monitor event counters                                                                                                                                                           | 0x06        |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xE00  | AMCFGR   | None  |

This interface is accessible as follows:

RO

## B.6.17 AMIIDR, Activity Monitors Implementation Identification Register

Defines the implementer and revisions of the AMU.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

**Component**  
AMU

**Register offset**  
0xE08

**Access type**  
RO

**Reset value**  
1101 1000 0101 0000 0010 0100 0011 1011

Bit descriptions

Figure B-86: ext\_amiidr bit assignments

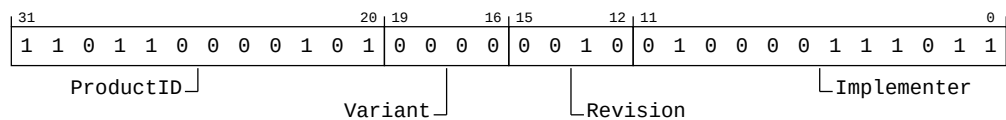


Table B-184: AMIIDR bit descriptions

| Bits    | Name        | Description                                                                                                                                                                | Reset  |
|---------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:20] | ProductID   | This field is an AMU part identifier.<br><br><b>0b110110000101</b><br>Cortex-X925                                                                                          | 0xD85  |
| [19:16] | Variant     | This field distinguishes product variants or major revisions of the product.<br><br><b>0b0000</b><br>rOp2                                                                  | 0b0000 |
| [15:12] | Revision    | This field distinguishes minor revisions of the product.<br><br><b>0b0010</b><br>rOp2                                                                                      | 0b0010 |
| [11:0]  | Implementer | Contains the JEP106 code of the company that implemented the AMU.<br><br>For an Arm implementation, this field reads as 0x43B.<br><br><b>0b010000111011</b><br>Arm Limited | 0x43B  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xE08  | AMIIDR   | None  |

This interface is accessible as follows:

RO



B.6.18 AMDEVARCH, Activity Monitors Device Architecture Register

Identifies the programmers' model architecture of the AMU component.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFBC

Access type

RO

Reset value

0100 0111 0111 0000 0000 1010 0110 0110

Bit descriptions

Figure B-87: ext\_amdevarch bit assignments

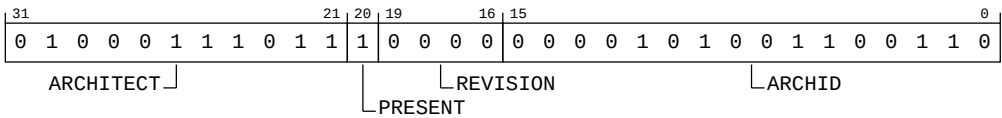


Table B-186: AMDEVARCH bit descriptions

| Bits    | Name      | Description                                                                                                                                         | Reset         |
|---------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [31:21] | ARCHITECT | Defines the architecture of the component. For AMU, this is Arm Limited.<br><b>0b01000111011</b>                                                    | 0b01000111011 |
| [20]    | PRESENT   | Indicates that the DEVARCH is present.<br><b>0b1</b>                                                                                                | 0b1           |
| [19:16] | REVISION  | Defines the architecture revision. For architectures defined by Arm this is the minor revision.<br><b>0b0000</b><br>Architecture revision is AMUv1. | 0b0000        |

| Bits   | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                     | Reset  |
|--------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [15:0] | ARCHID | <div>Defines this part to be an AMU component. For architectures defined by Arm this is further subdivided.</div> <div>For AMU:</div> <ul style="list-style-type: none"><li>Bits [15:12] are the architecture version, 0x0.</li><li>Bits [11:0] are the architecture part number, 0xA66.</li></ul> <div>This corresponds to AMU architecture version AMUv1.</div> <div>0b0000101001100110</div> | 0x0A66 |

Accessibility

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| AMU       | 0xFBC  | AMDEVARCH | None  |

This interface is accessible as follows:

RO

B.6.19 AMDEVTYPE, Activity Monitors Device Type Register

Indicates to a debugger that this component is part of a PE's performance monitor interface.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFCC

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0001 | 0110 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-88: ext\_amdevtype bit assignments



Table B-188: AMDEVTYPE bit descriptions

| Bits   | Name  | Description                                                                                                                   | Reset  |
|--------|-------|-------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                      | RES0   |
| [7:4]  | SUB   | Subtype. Reads as 0x1, to indicate this is a component within a PE.<br><br>0b0001<br>Component within a PE.                   | 0b0001 |
| [3:0]  | MAJOR | Major type. Reads as 0x6, to indicate this is a performance monitor component.<br><br>0b0110<br>Performance monitor component | 0b0110 |

Accessibility

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| AMU       | 0xFCC  | AMDEVTYPE | None  |

This interface is accessible as follows:

RO

B.6.20 AMPIDR4, Activity Monitors Peripheral Identification Register 4

Provides information to identify an activity monitors component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

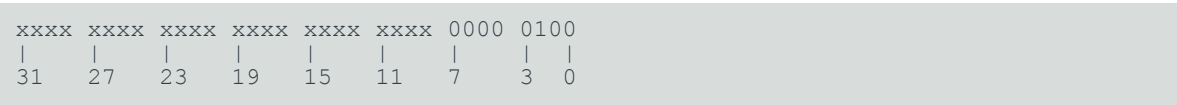
Register offset

0xFD0

Access type

RO

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-89: ext\_ampidr4 bit assignments



Table B-190: AMPIDR4 bit descriptions

| Bits   | Name  | Description                                                                                                                                                                                         | Reset  |
|--------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                                                                                            | RES0   |
| [7:4]  | SIZE  | Size of the component. Log <sub>2</sub> of the number of 4KB pages from the start of the component to the end of the component ID registers.<br><b>0b0000</b>                                       | 0b0000 |
| [3:0]  | DES_2 | Designer. JEP106 continuation code, least significant nibble.<br><br>For Arm Limited, this field is 0b0100.<br><b>0b0100</b><br><br>Arm Limited. This is bits[3:0] of the JEP106 continuation code. | 0b0100 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFD0  | AMPIDR4  | None  |

This interface is accessible as follows:

RO

B.6.21 AMPIDR0, Activity Monitors Peripheral Identification Register 0

Provides information to identify an activity monitors component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

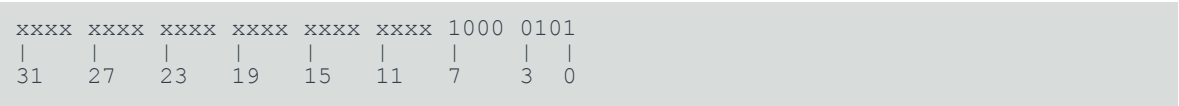
Register offset

0xFE0

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-90: ext\_ampidr0 bit assignments



Table B-192: AMPIDR0 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:8] | RES0 | Reserved    | RES0  |

| Bits  | Name   | Description                                                                                           | Reset |
|-------|--------|-------------------------------------------------------------------------------------------------------|-------|
| [7:0] | PART_0 | Part number, least significant byte.<br><br><b>0b10000101</b><br>Part number, least significant byte. | 0x85  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFE0  | AMPIDR0  | None  |

This interface is accessible as follows:

RO

B.6.22 AMPIDR1, Activity Monitors Peripheral Identification Register 1

Provides information to identify an activity monitors component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFE4

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1011 | 1101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-91: ext\_ampidr1 bit assignments

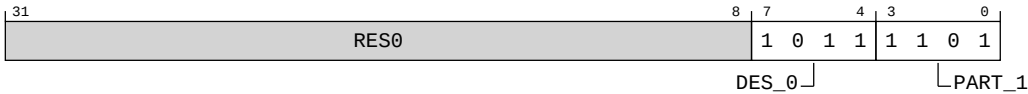


Table B-194: AMPIDR1 bit descriptions

| Bits   | Name   | Description                                                                                                                                                                       | Reset  |
|--------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0   | Reserved                                                                                                                                                                          | RES0   |
| [7:4]  | DES_0  | Designer, least significant nibble of JEP106 ID code.<br><br>For Arm Limited, this field is 0b1011.<br><br><b>0b1011</b><br>Designer, least significant nibble of JEP106 ID code. | 0b1011 |
| [3:0]  | PART_1 | Part number, most significant nibble.<br><br><b>0b1101</b><br>Part number, most significant nibble.                                                                               | 0b1101 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFE4  | AMPIDR1  | None  |

This interface is accessible as follows:

RO

B.6.23 AMPIDR2, Activity Monitors Peripheral Identification Register 2

Provides information to identify an activity monitors component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFE8

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-92: ext\_ampidr2 bit assignments

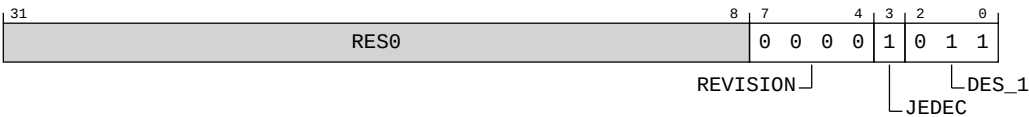


Table B-196: AMPIDR2 bit descriptions

| Bits   | Name     | Description                                                                                                                                                           | Reset  |
|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0     | Reserved                                                                                                                                                              | RES0   |
| [7:4]  | REVISION | Part major revision. Parts can also use this field to extend Part number to 16-bits.<br><br>0b0000<br>rOp2                                                            | 0b0000 |
| [3]    | JEDEC    | Indicates a JEP106 identity code is used.<br><br>0b1                                                                                                                  | 0b1    |
| [2:0]  | DES_1    | Designer, most significant bits of JEP106 ID code.<br><br>For Arm Limited, this field is 0b011.<br><br>0b011<br>Arm Limited. This is bits[6:4] of the JEP106 ID code. | 0b011  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFE8  | AMPIDR2  | None  |

This interface is accessible as follows:

RO



B.6.24 AMPIDR3, Activity Monitors Peripheral Identification Register 3

Provides information to identify an activity monitors component.

For more information, see *About the Peripheral identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

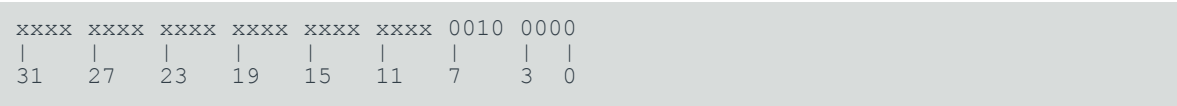
Register offset

0xFEC

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-93: ext\_ampidr3 bit assignments



Table B-198: AMPIDR3 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:8] | RES0 | Reserved    | RES0  |

| Bits  | Name   | Description                                                                                                                                                                  | Reset  |
|-------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [7:4] | REVAND | Part minor revision. Parts using ext-AMPIDR2.REVISION as an extension to the Part number must use this field as a major revision number.<br><br><b>0b0010</b>                | 0b0010 |
| [3:0] | CMOD   | Customer modified. Indicates someone other than the Designer has modified the component.<br><br><b>0b0000</b><br><br>The component is not modified from the original design. | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFEC  | AMPIDR3  | None  |

This interface is accessible as follows:

RO

B.6.25 AMCIDR0, Activity Monitors Component Identification Register 0

Provides information to identify an activity monitors component.

For more information, see *About the Component identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFF0

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-94: ext\_amcidr0 bit assignments

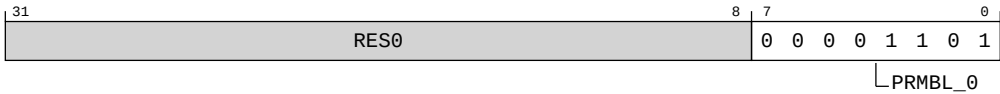


Table B-200: AMCIDR0 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_0 | Preamble.<br>0b00001101 | 0x0D  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFF0  | AMCIDR0  | None  |

This interface is accessible as follows:

RO

B.6.26 AMCIDR1, Activity Monitors Component Identification Register 1

Provides information to identify an activity monitors component.

For more information, see *About the Component identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFF4

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-95: ext\_amcldr1 bit assignments

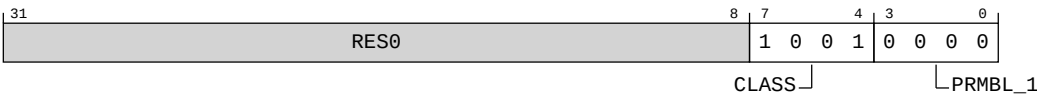


Table B-202: AMCIDR1 bit descriptions

| Bits   | Name    | Description                                               | Reset  |
|--------|---------|-----------------------------------------------------------|--------|
| [31:8] | RES0    | Reserved                                                  | RES0   |
| [7:4]  | CLASS   | Component class.<br><b>0b1001</b><br>CoreSight component. | 0b1001 |
| [3:0]  | PRMBL_1 | Preamble.<br><b>0b0000</b>                                | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFF4  | AMCIDR1  | None  |

This interface is accessible as follows:

RO

### B.6.27 AMCIDR2, Activity Monitors Component Identification Register 2

Provides information to identify an activity monitors component.

For more information, see *About the Component identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

AMU

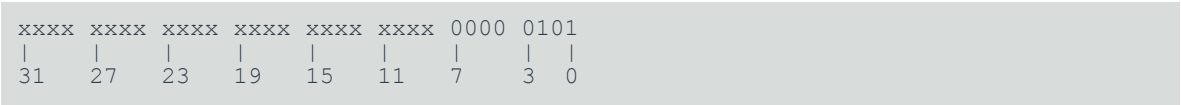
##### Register offset

0xFF8

##### Access type

RO

##### Reset value



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-96: ext\_amcidr2 bit assignments

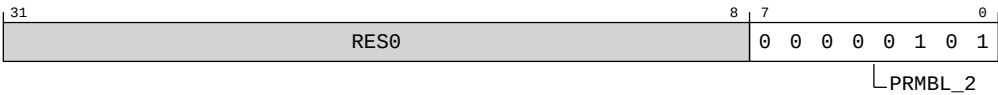


Table B-204: AMCIDR2 bit descriptions

| Bits   | Name    | Description             | Reset |
|--------|---------|-------------------------|-------|
| [31:8] | RES0    | Reserved                | RES0  |
| [7:0]  | PRMBL_2 | Preamble.<br>0b00000101 | 0x05  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFF8  | AMCIDR2  | None  |

This interface is accessible as follows:

RO

B.6.28 AMCIDR3, Activity Monitors Component Identification Register 3

Provides information to identify an activity monitors component.

For more information, see *About the Component identification scheme* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

AMU

Register offset

0xFFC

Access type

RO

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 1011 | 0001 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure B-97: ext\_amcidr3 bit assignments

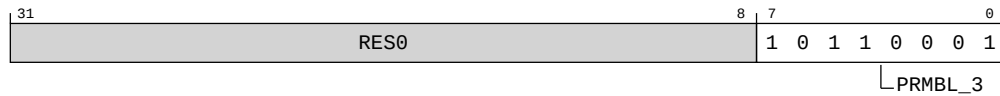


Table B-206: AMCIDR3 bit descriptions

| Bits   | Name    | Description                    | Reset |
|--------|---------|--------------------------------|-------|
| [31:8] | RES0    | Reserved                       | RES0  |
| [7:0]  | PRMBL_3 | Preamble.<br><b>0b10110001</b> | 0xB1  |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| AMU       | 0xFFC  | AMCIDR3  | None  |

This interface is accessible as follows:

RO

## B.7 External ETE registers summary

The following summary table provides an overview of all memory-mapped ETE registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

Table B-208: ETE registers summary

| Offset | Name                       | Reset                      | Width  | Description                |
|--------|----------------------------|----------------------------|--------|----------------------------|
| 0x018  | <a href="#">TRCAUXCTLR</a> | See individual bit resets. | 32-bit | Auxiliary Control Register |
| 0x180  | <a href="#">TRCIDR8</a>    | See individual bit resets. | 32-bit | ID Register 8              |
| 0x184  | <a href="#">TRCIDR9</a>    | See individual bit resets. | 32-bit | ID Register 9              |
| 0x188  | <a href="#">TRCIDR10</a>   | See individual bit resets. | 32-bit | ID Register 10             |
| 0x18C  | <a href="#">TRCIDR11</a>   | See individual bit resets. | 32-bit | ID Register 11             |
| 0x190  | <a href="#">TRCIDR12</a>   | See individual bit resets. | 32-bit | ID Register 12             |
| 0x194  | <a href="#">TRCIDR13</a>   | See individual bit resets. | 32-bit | ID Register 13             |
| 0x1C0  | <a href="#">TRCIMSPECO</a> | See individual bit resets. | 32-bit | IMP DEF Register 0         |

| Offset          | Name         | Reset                      | Width  | Description                                        |
|-----------------|--------------|----------------------------|--------|----------------------------------------------------|
| 0x1E0           | TRCIDR0      | See individual bit resets. | 32-bit | ID Register 0                                      |
| 0x1E4           | TRCIDR1      | See individual bit resets. | 32-bit | ID Register 1                                      |
| 0x1E8           | TRCIDR2      | See individual bit resets. | 32-bit | ID Register 2                                      |
| 0x1EC           | TRCIDR3      | See individual bit resets. | 32-bit | ID Register 3                                      |
| 0x1F0           | TRCIDR4      | See individual bit resets. | 32-bit | ID Register 4                                      |
| 0x1F4           | TRCIDR5      | See individual bit resets. | 32-bit | ID Register 5                                      |
| 0x1F8           | TRCIDR6      | See individual bit resets. | 32-bit | ID Register 6                                      |
| 0x1FC           | TRCIDR7      | See individual bit resets. | 32-bit | ID Register 7                                      |
| 0x2A0 + (4 * n) | TRCSSCSR_n_  | See individual bit resets. | 32-bit | Single-shot Comparator Control Status Register <n> |
| 0xF00           | TRCITCTRL    | See individual bit resets. | 32-bit | Integration Mode Control Register                  |
| 0xFA0           | TRCCCLAIMSET | See individual bit resets. | 32-bit | Claim Tag Set Register                             |
| 0xFA4           | TRCCCLAIMCLR | See individual bit resets. | 32-bit | Claim Tag Clear Register                           |
| 0xFBC           | TRCDEVARCH   | See individual bit resets. | 32-bit | Device Architecture Register                       |
| 0xFC0           | TRCDEVID2    | See individual bit resets. | 32-bit | Device Configuration Register 2                    |
| 0xFC4           | TRCDEVID1    | See individual bit resets. | 32-bit | Device Configuration Register 1                    |
| 0xFC8           | TRCDEVID     | See individual bit resets. | 32-bit | Device Configuration Register                      |
| 0xFCC           | TRCDEVTYPE   | See individual bit resets. | 32-bit | Device Type Register                               |
| 0xFD0           | TRCPIDR4     | See individual bit resets. | 32-bit | Peripheral Identification Register 4               |
| 0xFD4           | TRCPIDR5     | See individual bit resets. | 32-bit | Peripheral Identification Register 5               |
| 0xFD8           | TRCPIDR6     | See individual bit resets. | 32-bit | Peripheral Identification Register 6               |
| 0xFDC           | TRCPIDR7     | See individual bit resets. | 32-bit | Peripheral Identification Register 7               |
| 0xFE0           | TRCPIDR0     | See individual bit resets. | 32-bit | Peripheral Identification Register 0               |
| 0xFE4           | TRCPIDR1     | See individual bit resets. | 32-bit | Peripheral Identification Register 1               |
| 0xFE8           | TRCPIDR2     | See individual bit resets. | 32-bit | Peripheral Identification Register 2               |
| 0xFEC           | TRCPIDR3     | See individual bit resets. | 32-bit | Peripheral Identification Register 3               |
| 0xFF0           | TRCCIDR0     | See individual bit resets. | 32-bit | Component Identification Register 0                |
| 0xFF4           | TRCCIDR1     | See individual bit resets. | 32-bit | Component Identification Register 1                |
| 0xFF8           | TRCCIDR2     | See individual bit resets. | 32-bit | Component Identification Register 2                |
| 0xFFC           | TRCCIDR3     | See individual bit resets. | 32-bit | Component Identification Register 3                |

## B.7.1 TRCAUXCTRL, Auxiliary Control Register

The function of this register is **IMPLEMENTATION DEFINED**.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32



Component

ETE

Register offset

0x018

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-98: ext\_trcauxctlr bit assignments

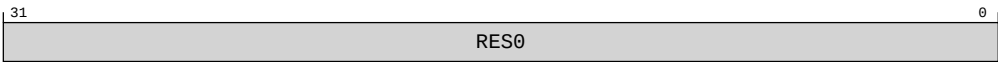


Table B-209: TRCAUXCTLR bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

If this register is nonzero then it might cause the behavior of a trace unit to contradict this architecture specification. See the documentation of the specific implementation for information about the IMPLEMENTATION DEFINED support for this register.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| ETE       | 0x018  | TRCAUXCTLR | None  |

This interface is accessible as follows:

When OSLockStatus() || !AllowExternalTraceAccess() || !IsTraceCorePowered()

ERROR

Otherwise

RW

B.7.2 TRCIDR8, ID Register 8

Returns the maximum speculation depth of the instruction trace element stream.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0x180

Access type

See bit descriptions

Reset value

0000 0000 0000 0000 0000 0000 0000 0000

Bit descriptions

Figure B-99: ext\_trcidr8 bit assignments

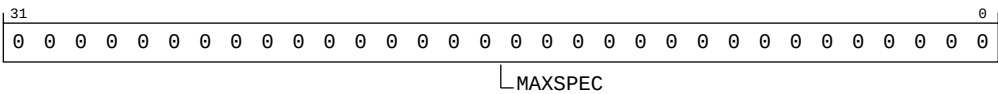


Table B-211: TRCIDR8 bit descriptions

| Bits   | Name    | Description                                                                                                                                                                                                                                                                         | Reset      |
|--------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:0] | MAXSPEC | Indicates the maximum speculation depth of the instruction trace element stream. This is the maximum number of PO elements in the trace element stream that can be speculative at any time.<br><br>0b00000000000000000000000000000000<br>No speculation in the trace element stream | 0x00000000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x180  | TRCIDR8  | None  |

This interface is accessible as follows:

When OSLockStatus() || !IsTraceCorePowered()  
ERROR

Otherwise  
RO

B.7.3 TRCIDR9, ID Register 9

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0x184

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-100: ext\_trcidr9 bit assignments



Table B-213: TRCIDR9 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x184  | TRCIDR9  | None  |

This interface is accessible as follows:

When OSLockStatus() || !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.4 TRCIDR10, ID Register 10

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width  
32

Component  
ETE

Register offset  
0x188

Access type  
See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-101: ext\_trcidr10 bit assignments

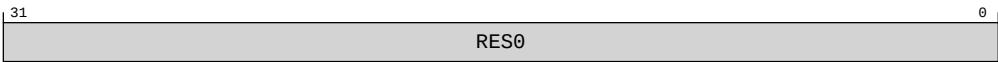


Table B-215: TRCIDR10 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x188  | TRCIDR10 | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**  
ERROR

**Otherwise**  
RO

B.7.5 TRCIDR11, ID Register 11

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

**Width**  
32

**Component**  
ETE

**Register offset**  
0x18C

**Access type**  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-102: ext\_trcidr11 bit assignments

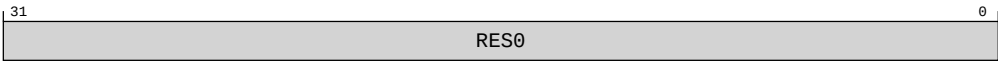


Table B-217: TRCIDR11 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x18C  | TRCIDR11 | None  |

This interface is accessible as follows:

When OSLockStatus() || !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.6 TRCIDR12, ID Register 12

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

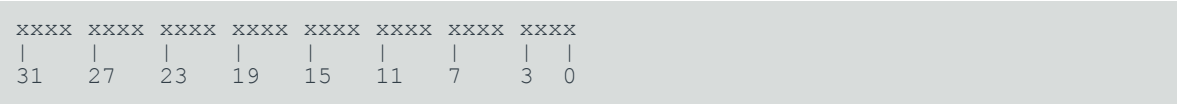
Register offset

0x190

Access type

See bit descriptions

Reset value





Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-103: ext\_trcldr12 bit assignments

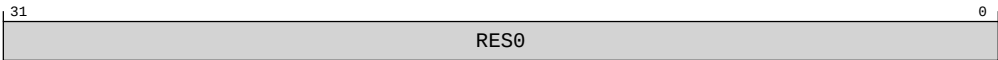


Table B-219: TRCIDR12 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x190  | TRCIDR12 | None  |

This interface is accessible as follows:

When `OSLockStatus() || !IsTraceCorePowered()`  
ERROR  
Otherwise  
RO

B.7.7 TRCIDR13, ID Register 13

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width  
32

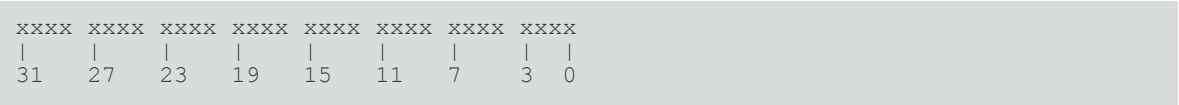
Component  
ETE

Register offset  
0x194

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-104: ext\_trcldr13 bit assignments

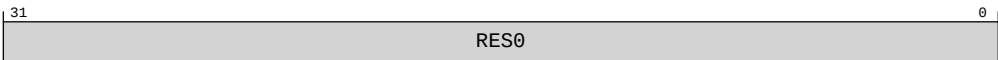


Table B-221: TRCIDR13 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x194  | TRCIDR13 | None  |

This interface is accessible as follows:

When OSLockStatus() || !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.8 TRCIMSPEC0, IMP DEF Register 0

TRCIMSPEC0 shows the presence of any **IMPLEMENTATION DEFINED** features, and provides an interface to enable the features that are provided.

Configurations

This register is available in all configurations.



Attributes

Width

32

Component

ETE

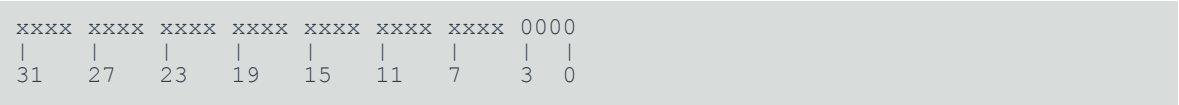
Register offset

0x1C0

Access type

RO

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-105: ext\_trcimspec0 bit assignments



Table B-223: TRCIMSPEC0 bit descriptions

| Bits   | Name    | Description                                                                                                                                                        | Reset  |
|--------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:4] | RES0    | Reserved                                                                                                                                                           | RES0   |
| [3:0]  | SUPPORT | Indicates whether the implementation supports <b>IMPLEMENTATION DEFINED</b> features.<br><b>0b0000</b><br>No <b>IMPLEMENTATION DEFINED</b> features are supported. | 0b0000 |

Accessibility

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| ETE       | 0x1C0  | TRCIMSPEC0 | None  |

This interface is accessible as follows:

```
When OSLockStatus() || !AllowExternalTraceAccess() || !IsTraceCorePowered()  
ERROR
```

Otherwise  
RW

B.7.9 TRCIDR0, ID Register 0

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width  
32

Component  
ETE

Register offset  
0x1E0

Access type  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| x010 | 1000 | xxxx | xxx0 | 00xx | 111x | 1010 | 000x |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-106: ext\_trcidr0 bit assignments

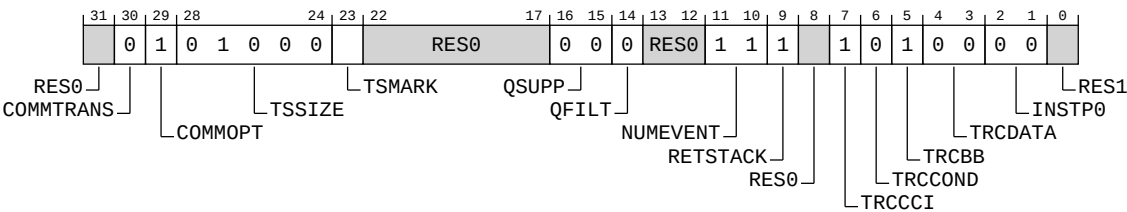


**Table B-225: TRCIDR0 bit descriptions**

| Bits    | Name      | Description                                                                                                                                                                                                                                                       | Reset                                                                     |
|---------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| [31]    | RES0      | Reserved                                                                                                                                                                                                                                                          | RES0                                                                      |
| [30]    | COMMTRANS | Transaction Start element behavior.<br><br><b>0b0</b><br>Transaction Start elements are P0 elements.                                                                                                                                                              | 0b0                                                                       |
| [29]    | COMMOPT   | Indicates the contents and encodings of Cycle count packets.<br><br><b>0b1</b><br>Commit mode 1.<br><br>Access to this field is: <b>RAO/WI</b>                                                                                                                    | 0b1                                                                       |
| [28:24] | TSSIZE    | Indicates that the trace unit implements Global timestamping and the size of the timestamp value.<br><br><b>0b01000</b><br>Global timestamping implemented with a 64-bit timestamp value.                                                                         | 0b01000                                                                   |
| [23]    | TSMARK    | Indicates whether Timestamp Marker elements are generated.<br><br><b>0b0</b><br>Timestamp Marker elements are not generated.<br><br><b>0b1</b><br>Timestamp Marker elements are generated.                                                                        | The reset values can be the following: 0b0, 0b1, respective to the value. |
| [22:17] | RES0      | Reserved                                                                                                                                                                                                                                                          | RES0                                                                      |
| [16:15] | QSUPP     | Indicates that the trace unit implements Q element support.<br><br><b>0b00</b><br>Q element support is not implemented.                                                                                                                                           | 0b00                                                                      |
| [14]    | QFILT     | Indicates if the trace unit implements Q element filtering.<br><br><b>0b0</b><br>Q element filtering is not implemented.                                                                                                                                          | 0b0                                                                       |
| [13:12] | RES0      | Reserved                                                                                                                                                                                                                                                          | RES0                                                                      |
| [11:10] | NUMEVENT  | Indicates the number of ETEEvents implemented.<br><br><b>0b11</b><br>The trace unit supports 4 ETEEvents.                                                                                                                                                         | 0b11                                                                      |
| [9]     | RETSTACK  | Indicates if the trace unit supports the return stack.<br><br><b>0b1</b><br>Return stack implemented.                                                                                                                                                             | 0b1                                                                       |
| [8]     | RES0      | Reserved                                                                                                                                                                                                                                                          | RES0                                                                      |
| [7]     | TRCCCI    | Indicates if the trace unit implements cycle counting.<br><br><b>0b1</b><br>Cycle counting implemented.                                                                                                                                                           | 0b1                                                                       |
| [6]     | TRCCOND   | Indicates if the trace unit implements conditional instruction tracing. Conditional instruction tracing is not implemented in ETE and this field is reserved for other trace architectures.<br><br><b>0b0</b><br>Conditional instruction tracing not implemented. | 0b0                                                                       |

| Bits  | Name    | Description                                                                                                                                                                                                                                                                 | Reset |
|-------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [5]   | TRCBB   | Indicates if the trace unit implements branch broadcasting.<br><br><b>0b1</b><br>Branch broadcasting implemented.                                                                                                                                                           | 0b1   |
| [4:3] | TRCDATA | Indicates if the trace unit implements data tracing. Data tracing is not implemented in ETE and this field is reserved for other trace architectures.<br><br><b>0b00</b><br>Data tracing not implemented.                                                                   | 0b00  |
| [2:1] | INSTPO  | Indicates if load and store instructions are PO instructions. Load and store instructions as PO instructions is not implemented in ETE and this field is reserved for other trace architectures.<br><br><b>0b00</b><br>Load and store instructions are not PO instructions. | 0b00  |
| [0]   | RES1    | Reserved                                                                                                                                                                                                                                                                    | RES1  |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1E0  | TRCIDR0  | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

## B.7.10 TRCIDR1, ID Register 1

Returns the tracing capabilities of the trace unit.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Component

ETE

#### Register offset

0x1E4

#### Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0100 | 0001 | xxxx | xxxx | xxxx | 1111 | 1111 | 0000 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      | 0    |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-107: ext\_trcidr1 bit assignments

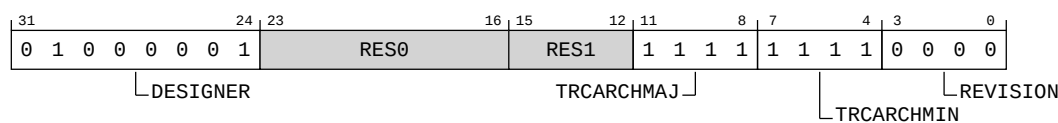


Table B-227: TRCIDR1 bit descriptions

| Bits    | Name       | Description                                                                                                                                                               | Reset  |
|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:24] | DESIGNER   | Indicates which company designed the trace unit. The permitted values of this field are the same as AArch64-MIDR_EL1.Implementer.<br><br><b>0b01000001</b><br>Arm Limited | 0x41   |
| [23:16] | RES0       | Reserved                                                                                                                                                                  | RES0   |
| [15:12] | RES1       | Reserved                                                                                                                                                                  | RES1   |
| [11:8]  | TRCARCHMAJ | Major architecture version.<br><br><b>0b1111</b><br>If both TRCARCHMAJ and TRCARCHMIN == 0xF then refer to ext-TRCDEVARCH.                                                | 0b1111 |
| [7:4]   | TRCARCHMIN | Minor architecture version.<br><br><b>0b1111</b><br>If both TRCARCHMAJ and TRCARCHMIN == 0xF then refer to ext-TRCDEVARCH.                                                | 0b1111 |
| [3:0]   | REVISION   | Implementation revision that identifies the revision of the trace and OS Lock registers.<br><br><b>0b0000</b><br>Revision 0                                               | 0b0000 |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1E4  | TRCIDR1  | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**

ERROR

Otherwise  
RO

B.7.11 TRCIDR2, ID Register 2

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

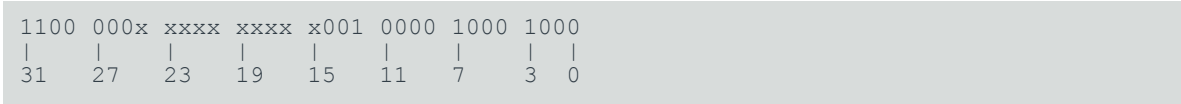
Width  
32

Component  
ETE

Register offset  
0x1E8

Access type  
See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-108: ext\_trcidr2 bit assignments

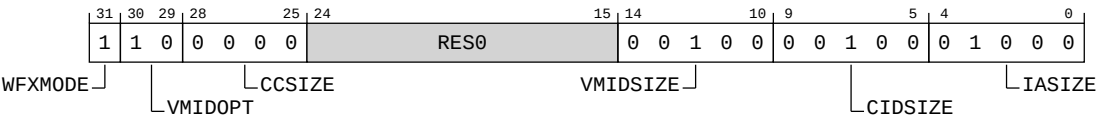


Table B-229: TRCIDR2 bit descriptions

| Bits | Name    | Description                                                                                                                                                                           | Reset |
|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | WFXMODE | Indicates whether WFI, WFIT, WFE, and WFET instructions are classified as PO instructions:<br><br>0b1<br><br>WFI, WFIT, WFE, and WFET instructions are classified as PO instructions. | 0b1   |

| Bits    | Name     | Description                                                                                                                                                                   | Reset   |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [30:29] | VMIDOPT  | Indicates the options for Virtual context identifier selection.<br><b>0b10</b><br>Virtual context identifier selection not supported. ext-TRCCONFIGR.VMIDOPT is <b>RES1</b> . | 0b10    |
| [28:25] | CCSIZE   | Indicates the size of the cycle counter.<br><b>0b0000</b><br>The cycle counter is 12 bits in length.                                                                          | 0b0000  |
| [24:15] | RES0     | Reserved                                                                                                                                                                      | RES0    |
| [14:10] | VMIDSIZE | Indicates the trace unit Virtual context identifier size.<br><b>0b00100</b><br>32-bit Virtual context identifier size.                                                        | 0b00100 |
| [9:5]   | CIDSIZE  | Indicates the Context identifier size.<br><b>0b00100</b><br>32-bit Context identifier size.                                                                                   | 0b00100 |
| [4:0]   | IASIZE   | Virtual instruction address size.<br><b>0b01000</b><br>Maximum of 64-bit instruction address size.                                                                            | 0b01000 |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1E8  | TRCIDR2  | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

## B.7.12 TRCIDR3, ID Register 3

Returns the base architecture of the trace unit.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Component

ETE

**Register offset**

0x1EC

**Access type**

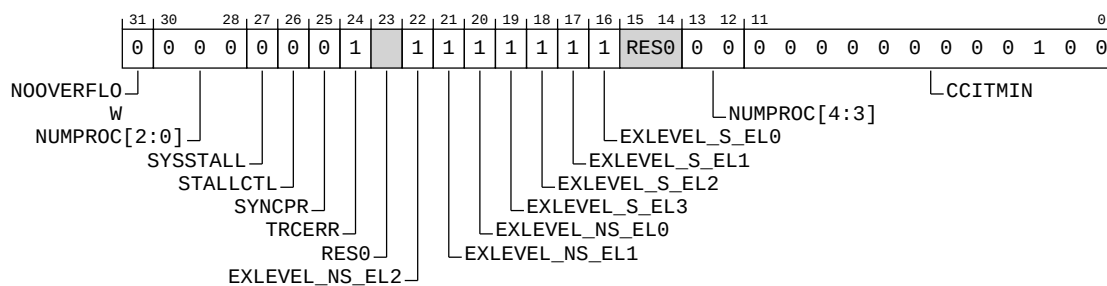
See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0000 | 0001 | x111 | 1111 | xx00 | 0000 | 0000 | 0100 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |

**Note**

Where the reset reads xxxx, see individual bits.

**Bit descriptions****Figure B-109: ext\_trcidr3 bit assignments****Table B-231: TRCIDR3 bit descriptions**

| Bits | Name       | Description                                                                                                                                                        | Reset |
|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | NOOVERFLOW | Indicates if overflow prevention is implemented.<br><b>0b0</b><br>Overflow prevention is not implemented.                                                          | 0b0   |
| [27] | SYSSTALL   | Indicates if stalling of the PE is permitted.<br><b>0b0</b><br>Stalling of the PE is not permitted.                                                                | 0b0   |
| [26] | STALLCTL   | Indicates if trace unit implements stalling of the PE.<br><b>0b0</b><br>Stalling of the PE is not implemented.                                                     | 0b0   |
| [25] | SYNCPR     | Indicates if an implementation has a fixed synchronization period.<br><b>0b0</b><br>ext-TRCSYNCPR is read/write so software can change the synchronization period. | 0b0   |



| Bits           | Name           | Description                                                                                                                                                                                                                                         | Reset   |
|----------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [24]           | TRCERR         | Indicates forced tracing of System Error exceptions is implemented.<br><b>0b1</b><br>Forced tracing of System Error exceptions is implemented.                                                                                                      | 0b1     |
| [23]           | RES0           | Reserved                                                                                                                                                                                                                                            | RES0    |
| [22]           | EXLEVEL_NS_EL2 | Indicates if Non-secure EL2 implemented.<br><b>0b1</b><br>Non-secure EL2 is implemented.                                                                                                                                                            | 0b1     |
| [21]           | EXLEVEL_NS_EL1 | Indicates if Non-secure EL1 implemented.<br><b>0b1</b><br>Non-secure EL1 is implemented.                                                                                                                                                            | 0b1     |
| [20]           | EXLEVEL_NS_ELO | Indicates if Non-secure ELO implemented.<br><b>0b1</b><br>Non-secure ELO is implemented.                                                                                                                                                            | 0b1     |
| [19]           | EXLEVEL_S_EL3  | Indicates if Secure EL3 implemented.<br><b>0b1</b><br>EL3 is implemented.                                                                                                                                                                           | 0b1     |
| [18]           | EXLEVEL_S_EL2  | Indicates if Secure EL2 implemented.<br><b>0b1</b><br>Secure EL2 is implemented.                                                                                                                                                                    | 0b1     |
| [17]           | EXLEVEL_S_EL1  | Indicates if Secure EL1 implemented.<br><b>0b1</b><br>Secure EL1 is implemented.                                                                                                                                                                    | 0b1     |
| [16]           | EXLEVEL_S_ELO  | Indicates if Secure ELO implemented.<br><b>0b1</b><br>Secure ELO is implemented.                                                                                                                                                                    | 0b1     |
| [15:14]        | RES0           | Reserved                                                                                                                                                                                                                                            | RES0    |
| [13:12, 30:28] | NUMPROC        | Indicates the number of PEs available for tracing.<br><b>0b00000</b><br>The trace unit can trace one PE.                                                                                                                                            | 0b00000 |
| [11:0]         | CCITMIN        | Indicates the minimum value that can be programmed in ext-TRCCCCTLR.THRESHOLD.<br><br>If ext-TRCIDR0.TRCCCI == 1 then the minimum value of this field is 0x001.<br><br>If ext-TRCIDR0.TRCCCI == 0 then this field is zero.<br><b>0b000000000100</b> | 0x004   |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1EC  | TRCIDR3  | None  |

This interface is accessible as follows:

When OSLockStatus() || !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.13 TRCIDR4, ID Register 4

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0x1F0

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0001 | 0001 | 0001 | 0111 | 0000 | xxx0 | 0000 | 0100 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-110: ext\_trcidr4 bit assignments

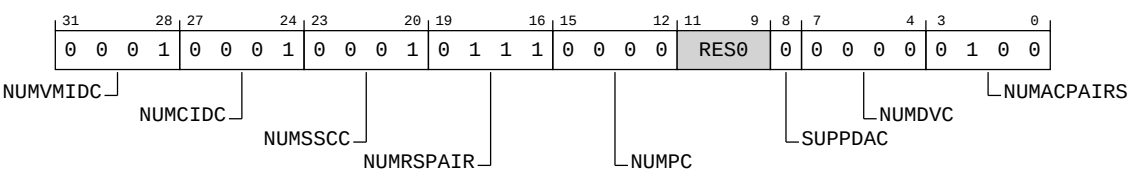


**Table B-233: TRCIDR4 bit descriptions**

| Bits    | Name        | Description                                                                                                                                                                                                                                                            | Reset       |
|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [31:28] | NUMVMIDC    | Indicates the number of Virtual Context Identifier Comparators that are available for tracing.<br><br><b>0b0001</b><br>The implementation has one Virtual Context Identifier Comparator.                                                                               | 0b0001      |
| [27:24] | NUMCIDC     | Indicates the number of Context Identifier Comparators that are available for tracing.<br><br><b>0b0001</b><br>The implementation has one Context Identifier Comparator.                                                                                               | 0b0001      |
| [23:20] | NUMSSCC     | Indicates the number of Single-shot Comparator Controls that are available for tracing.<br><br><b>0b0001</b><br>The implementation has one Single-shot Comparator Control.                                                                                             | 0b0001      |
| [19:16] | NUMRSPAIR   | Indicates the number of resource selector pairs that are available for tracing.<br><br><b>0b0111</b><br>The implementation has eight resource selector pairs.                                                                                                          | 0b0111      |
| [15:12] | NUMPC       | Indicates the number of PE Comparator Inputs that are available for tracing.<br><br><b>0b0000</b><br>No PE Comparator Inputs are available.                                                                                                                            | 0b0000      |
| [11:9]  | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                               | <b>RES0</b> |
| [8]     | SUPPDAC     | Indicates whether data address comparisons are implemented. Data address comparisons are not implemented in ETE and are reserved for other trace architectures. Allocated in other trace architectures.<br><br><b>0b0</b><br>Data address comparisons not implemented. | 0b0         |
| [7:4]   | NUMDVC      | Indicates the number of data value comparators. Data value comparators are not implemented in ETE and are reserved for other trace architectures. Allocated in other trace architectures.<br><br><b>0b0000</b><br>No data value comparators implemented.               | 0b0000      |
| [3:0]   | NUMACPAIRS  | Indicates the number of Address Comparator pairs that are available for tracing.<br><br><b>0b0100</b><br>The implementation has four Address Comparator pairs.                                                                                                         | 0b0100      |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1F0  | TRCIDR4  | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

B.7.14 TRCIDR5, ID Register 5

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0x1F4

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 0010 | 100x | 0100 | 0111 | xxxx | 1001 | 1111 | 1111 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-111: ext\_trcidr5 bit assignments

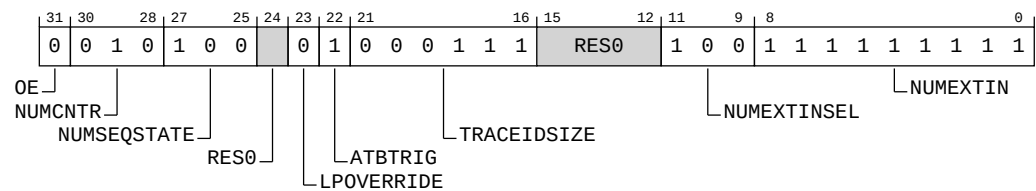


Table B-235: TRCIDR5 bit descriptions

| Bits | Name | Description                                                                                                             | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | OE   | Indicates support for the ETE Trace Output Enable.<br><br><b>0b0</b><br><br>ETE Trace Output Enable is not implemented. | 0b0   |

| Bits    | Name        | Description                                                                                                                                                      | Reset      |
|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [30:28] | NUMCNTR     | Indicates the number of Counters that are available for tracing.<br><br><b>0b010</b><br>Two Counters implemented.                                                | 0b010      |
| [27:25] | NUMSEQSTATE | Indicates if the Sequencer is implemented and the number of Sequencer states that are implemented.<br><br><b>0b100</b><br>Four Sequencer states are implemented. | 0b100      |
| [24]    | RES0        | Reserved                                                                                                                                                         | RES0       |
| [23]    | LPOVERRIDE  | Indicates support for Low-power Override Mode.<br><br><b>0b0</b><br>The trace unit does not support Low-power Override Mode.                                     | 0b0        |
| [22]    | ATBTRIG     | Indicates if the implementation can support ATB triggers.<br><br><b>0b1</b><br>The implementation supports ATB triggers.                                         | 0b1        |
| [21:16] | TRACEIDSIZE | Indicates the trace ID width.<br><br><b>0b000111</b><br>The implementation supports a 7-bit trace ID.                                                            | 0b000111   |
| [15:12] | RES0        | Reserved                                                                                                                                                         | RES0       |
| [11:9]  | NUMEXTINSEL | Indicates how many External Input Selector resources are implemented.<br><br><b>0b100</b><br>4 External Input Selector resources are available.                  | 0b100      |
| [8:0]   | NUMEXTIN    | Indicates how many External Inputs are implemented.<br><br><b>0b11111111</b><br>Unified PMU event selection.                                                     | 0b11111111 |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1F4  | TRCIDR5  | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

## B.7.15 TRCIDR6, ID Register 6

Returns the tracing capabilities of the trace unit.

### Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0x1F8

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-112: ext\_trcidr6 bit assignments

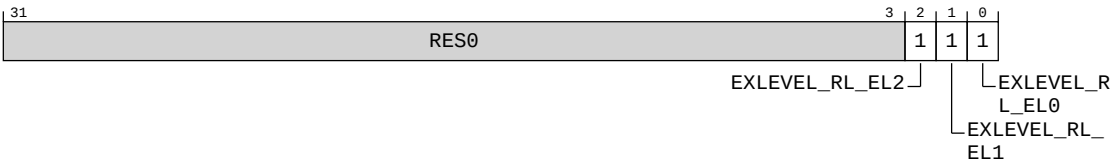


Table B-237: TRCIDR6 bit descriptions

| Bits   | Name           | Description                                                                       | Reset |
|--------|----------------|-----------------------------------------------------------------------------------|-------|
| [31:3] | RES0           | Reserved                                                                          | RES0  |
| [2]    | EXLEVEL_RL_EL2 | Indicates if Realm EL2 is implemented.<br><b>0b1</b><br>Realm EL2 is implemented. | 0b1   |
| [1]    | EXLEVEL_RL_EL1 | Indicates if Realm EL1 is implemented.<br><b>0b1</b><br>Realm EL1 is implemented. | 0b1   |
| [0]    | EXLEVEL_RL_ELO | Indicates if Realm ELO is implemented.<br><b>0b1</b><br>Realm ELO is implemented. | 0b1   |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1F8  | TRCIDR6  | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**  
ERROR

**Otherwise**  
RO

B.7.16 TRCIDR7, ID Register 7

Returns the tracing capabilities of the trace unit.

Configurations

This register is available in all configurations.

Attributes

**Width**  
32

**Component**  
ETE

**Register offset**  
0x1FC

**Access type**  
See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-113: ext\_trcidr7 bit assignments

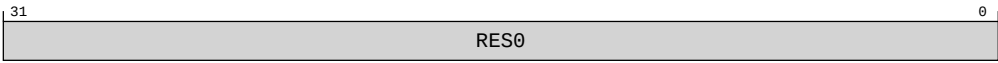


Table B-239: TRCIDR7 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0x1FC  | TRCIDR7  | None  |

This interface is accessible as follows:

When OSLockStatus() || !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.17 TRCSSCSR<n>, Single-shot Comparator Control Status Register  
<n> , n = 0 - 7

Returns the status of the corresponding Single-shot Comparator Control.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

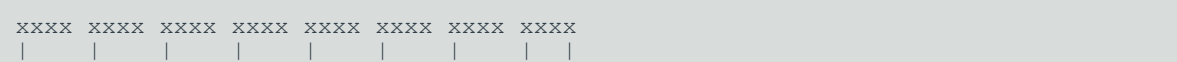
Register offset

0x2A0 + (4 \* n)

Access type

RO

Reset value





31 27 23 19 15 11 7 3 0



Note

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure B-114: ext\_trcsscsr\_n\_ bit assignments**

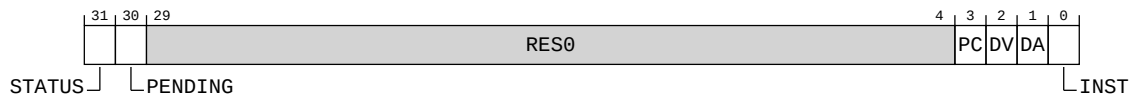


**Table B-241: TRCSSCSR<n> bit descriptions**

| Bits   | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|--------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31]   | STATUS  | <p>Single-shot Comparator Control status. Indicates if any of the comparators selected by this Single-shot Comparator control have matched. The selected comparators are defined by ext-TRCSSCCR&lt;n&gt;.ARC, ext-TRCSSCCR&lt;n&gt;.SAC, and ext-TRCSSPCICR&lt;n&gt;.PC.</p> <p><b>0b0</b></p> <p>No match has occurred. When the first match occurs, this field takes a value of 1. It remains at 1 until explicitly modified by a write to this register.</p> <p><b>0b1</b></p> <p>One or more matches has occurred. If ext-TRCSSCCR&lt;n&gt;.RST == 0 then:</p> <ul style="list-style-type: none"> <li>There is only one match and no more matches are possible.</li> <li>Software must reset this field to 0 to re-enable the Single-shot Comparator Control.</li> </ul> | x     |
| [30]   | PENDING | <p>Single-shot pending status. The Single-shot Comparator Control fired while the resources were in the Paused state.</p> <p><b>0b0</b></p> <p>No match has occurred.</p> <p><b>0b1</b></p> <p>One or more matches has occurred.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | x     |
| [29:4] | RES0    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0  |
| [3]    | PC      | <p>PE Comparator Input support. Indicates if the Single-shot Comparator Control supports PE Comparator Inputs.</p> <p><b>0b0</b></p> <p>This Single-shot Comparator Control does not support PE Comparator Inputs. Selecting any PE Comparator Inputs using the associated ext-TRCSSPCICR&lt;n&gt; results in <b>CONSTRAINED UNPREDICTABLE</b> behavior of the Single-shot Comparator Control resource. The Single-shot Comparator Control might match unexpectedly or might not match.</p> <p><b>0b1</b></p> <p>This Single-shot Comparator Control supports PE Comparator Inputs.</p>                                                                                                                                                                                       | x     |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [2]  | DV   | <p>Data value comparator support. Data value comparisons are not implemented in ETE and are reserved for other trace architectures. Allocated in other trace architectures.</p> <p><b>0b0</b><br/>This Single-shot Comparator Control does not support data value comparisons.</p> <p><b>0b1</b><br/>This Single-shot Comparator Control supports data value comparisons.</p>         | x     |
| [1]  | DA   | <p>Data Address Comparator support. Data address comparisons are not implemented in ETE and are reserved for other trace architectures. Allocated in other trace architectures.</p> <p><b>0b0</b><br/>This Single-shot Comparator Control does not support data address comparisons.</p> <p><b>0b1</b><br/>This Single-shot Comparator Control supports data address comparisons.</p> | x     |
| [0]  | INST | <p>Instruction Address Comparator support. Indicates if the Single-shot Comparator Control supports instruction address comparisons.</p> <p><b>0b0</b><br/>This Single-shot Comparator Control does not support instruction address comparisons.</p> <p><b>0b1</b><br/>This Single-shot Comparator Control supports instruction address comparisons.</p>                              | x     |

### Access

Must be programmed if TRCRSCTLR<a>.GROUP == 0b0011 and TRCRSCTLR<a>.SINGLE\_SHOT[n] == 1.

Writes are **CONSTRAINED UNPREDICTABLE** if the trace unit is not in the Idle state.

Reads from this register might return an **UNKNOWN** value if the trace unit is not in either of the Idle or Stable states.

### Accessibility

Must be programmed if TRCRSCTLR<a>.GROUP == 0b0011 and TRCRSCTLR<a>.SINGLE\_SHOT[n] == 1.

Writes are **CONSTRAINED UNPREDICTABLE** if the trace unit is not in the Idle state.

Reads from this register might return an **UNKNOWN** value if the trace unit is not in either of the Idle or Stable states.

| Component | Offset          | Instance    | Range |
|-----------|-----------------|-------------|-------|
| ETE       | 0x2A0 + (4 * n) | TRCSSCSR<n> | None  |

This interface is accessible as follows:

**When OSLockStatus() || !AllowExternalTraceAccess() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RW

B.7.18 TRCITCTRL, Integration Mode Control Register

A component can use TRCITCTRL to dynamically switch between functional mode and integration mode. In integration mode, topology detection is enabled. After switching to integration mode and performing integration tests or topology detection, reset the system to ensure correct behavior of CoreSight and other connected system components.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xF00

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-115: ext\_trcitctrl bit assignments

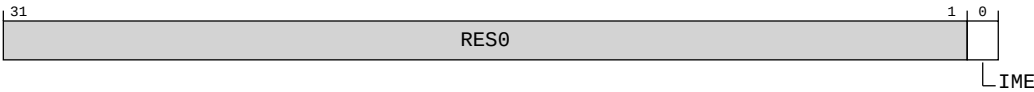


Table B-243: TRCITCTRL bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:1] | RES0 | Reserved    | RES0  |

| Bits | Name | Description                                                                                                                                                                                  | Reset |
|------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | IME  | Integration Mode Enable.<br><br><b>0b0</b><br>Component functional mode.<br><br><b>0b1</b><br>Component integration mode. Support for topology detection and integration testing is enabled. | x     |

### Accessibility

External debugger accesses to this register are IMPLEMENTATION DEFINED when the trace unit is not in the Idle state.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| ETE       | 0xF00  | TRCITCTRL | None  |

This interface is accessible as follows:

**When OSLockStatus() || !AllowExternalTraceAccess() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RW

## B.7.19 TRCCLAIMSET, Claim Tag Set Register

In conjunction with ext-TRCCLAIMCLR, provides Claim Tag bits that can be separately set and cleared to indicate whether functionality is in use by a debug agent.

For additional information, see the CoreSight Architecture Specification.

### Configurations

The number of claim tag bits implemented is IMPLEMENTATION DEFINED. Arm recommends that implementations support a minimum of four claim tag bits, that is, SET[3:0] reads as 0b1111.

### Attributes

#### Width

32

#### Component

ETE

#### Register offset

0xFA0

#### Access type

RAOW1S

#### Reset value

1111 1111 1111 1111 1111 1111 1111 1111

Bit descriptions

Figure B-116: ext\_trcclaimset bit assignments

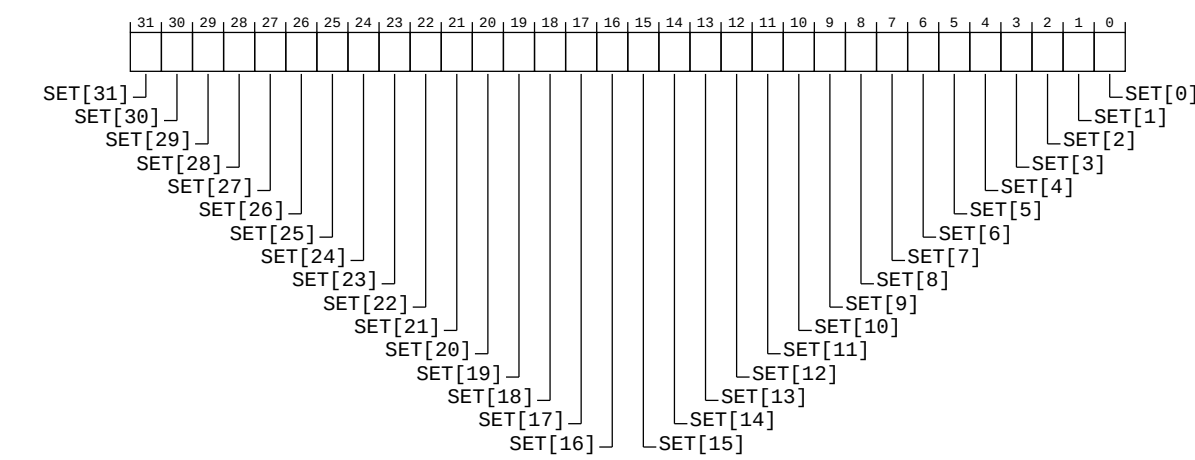


Table B-245: TRCCLAIMSET bit descriptions

| Bits   | Name                                | Description                                                                                                                                                                                                                                                                                                                                                                   | Reset      |
|--------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:0] | SET[<m>], bit[m], where m = 31 to 0 | <div>Claim Tag Set. Indicates whether Claim Tag bit &lt;m&gt; is implemented, and is used to set Claim Tag bit &lt;m&gt; to 1.</div> <div><b>0b0</b><br/>On a read: Claim Tag bit &lt;m&gt; is not implemented.<br/>On a write: Ignored.</div> <div><b>0b1</b><br/>On a read: Claim Tag bit &lt;m&gt; is implemented.<br/>On a write: Set Claim Tag bit &lt;m&gt; to 1.</div> | 0xFFFFFFFF |

Accessibility

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| ETE       | 0xFA0  | TRCCLAIMSET | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**  
ERROR

**Otherwise**  
RW

B.7.20 TRCCLAIMCLR, Claim Tag Clear Register

In conjunction with ext-TRCCLAIMSET, provides Claim Tag bits that can be separately set and cleared to indicate whether functionality is in use by a debug agent.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xFA4

Access type

RW1C

Reset value

0000 0000 0000 0000 0000 0000 0000 0000

Bit descriptions

Figure B-117: ext\_trclaimclr bit assignments

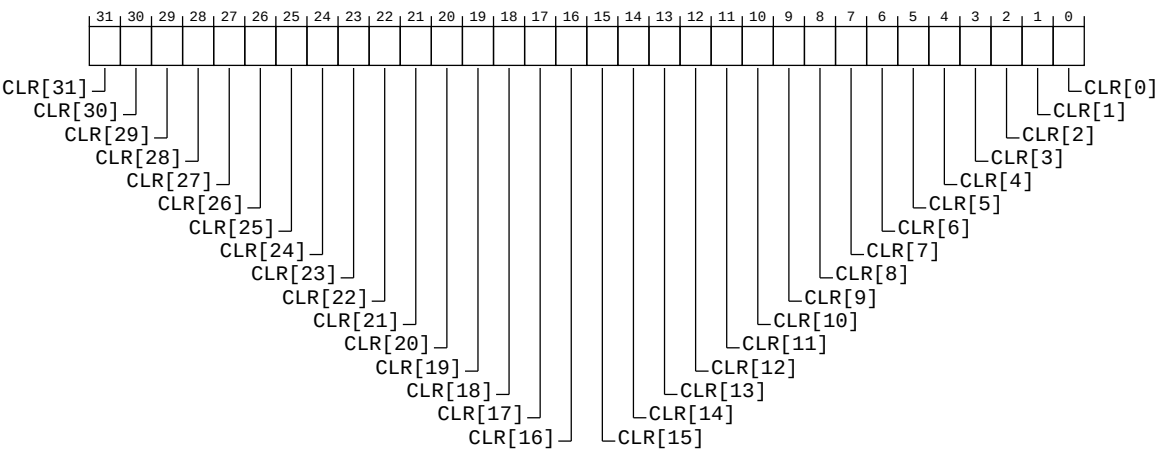


**Table B-247: TRCCLAIMCLR bit descriptions**

| Bits   | Name                                | Description                                                                                                                                                                                                                                                                                                                                                        | Reset      |
|--------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| [31:0] | CLR[<m>], bit[m], where m = 31 to 0 | <p>Claim Tag Clear. Indicates the current status of Claim Tag bit &lt;m&gt;, and is used to clear Claim Tag bit &lt;m&gt; to 0.</p> <p><b>0b0</b></p> <p>On a read: Claim Tag bit &lt;m&gt; is not set.</p> <p>On a write: Ignored.</p> <p><b>0b1</b></p> <p>On a read: Claim Tag bit &lt;m&gt; is set.</p> <p>On a write: Clear Claim tag bit &lt;m&gt; to 0.</p> | 0x00000000 |

### Accessibility

| Component | Offset | Instance    | Range |
|-----------|--------|-------------|-------|
| ETE       | 0xFA4  | TRCCLAIMCLR | None  |

This interface is accessible as follows:

**When OSLockStatus() || !IsTraceCorePowered()**

ERROR

**Otherwise**

RW

## B.7.21 TRCDEVARCH, Device Architecture Register

Provides discovery information for the component.

For additional information, see the CoreSight Architecture Specification.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Component

ETE

#### Register offset

0xFBC

#### Access type

See bit descriptions

Reset value

0100 0111 0111 0001 0101 1010 0001 0011

Bit descriptions

Figure B-118: ext\_trcdevarch bit assignments

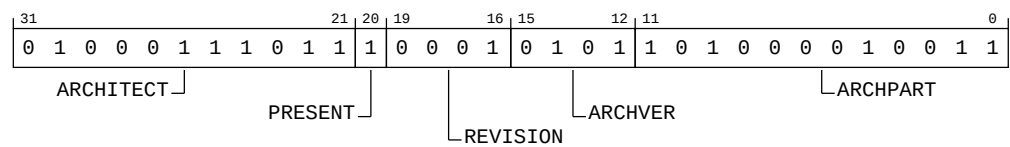


Table B-249: TRCDEVARCH bit descriptions

| Bits    | Name      | Description                                                                                                                                                                                                                                    | Reset         |
|---------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| [31:21] | ARCHITECT | Architect. Defines the architect of the component. Bits [31:28] are the JEP106 continuation code (JEP106 bank ID, minus 1) and bits [27:21] are the JEP106 ID code.<br><br><b>0b01000111011</b><br>JEP106 continuation code 0x4, ID code 0x3B. | 0b01000111011 |
| [20]    | PRESENT   | DEVARCH Present. Defines that the DEVARCH register is present.<br><br><b>0b1</b><br>Device Architecture information present.                                                                                                                   | 0b1           |
| [19:16] | REVISION  | Revision. Defines the architecture revision of the component. Defined values are:<br><br><b>0b0001</b><br>ETEv1.1, FEAT_ETEv1p1.                                                                                                               | 0b0001        |
| [15:12] | ARCHVER   | Architecture Version. Defines the architecture version of the component.<br><br><b>0b0101</b><br>ETEv1.                                                                                                                                        | 0b0101        |
| [11:0]  | ARCHPART  | Architecture Part. Defines the architecture of the component.<br><br><b>0b101000010011</b><br>Arm PE trace architecture.                                                                                                                       | 0xA13         |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| ETE       | 0xFBC  | TRCDEVARCH | None  |

This interface is accessible as follows:

**When !IsTraceCorePowered()**  
ERROR

**Otherwise**  
RO



B.7.22 TRCDEVID2, Device Configuration Register 2

Provides discovery information for the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

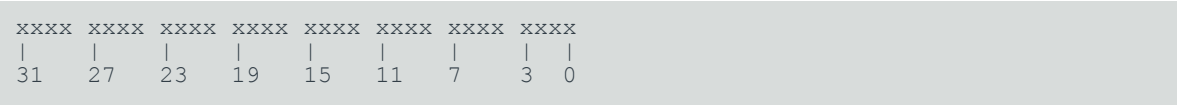
Register offset

0xFC0

Access type

See bit descriptions

Reset value



Note

Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-119: ext\_trcdevid2 bit assignments



Table B-251: TRCDEVID2 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| ETE       | 0xFC0  | TRCDEVID2 | None  |

This interface is accessible as follows:

**When !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

### B.7.23 TRCDEVID1, Device Configuration Register 1

Provides discovery information for the component.

For additional information, see the CoreSight Architecture Specification.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

ETE

##### Register offset

0xFC4

##### Access type

See bit descriptions

##### Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-120: ext\_trcdevid1 bit assignments

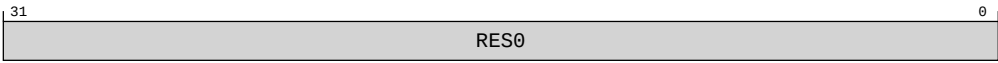


Table B-253: TRCDEVID1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| ETE       | 0xFC4  | TRCDEVID1 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.24 TRCDEVID, Device Configuration Register

Provides discovery information for the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

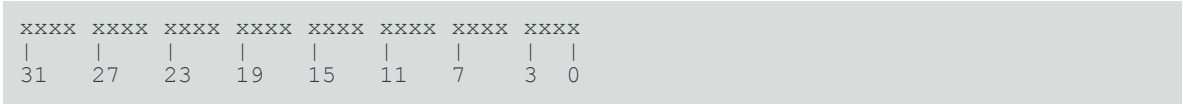
Register offset

0xFC8

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-121: ext\_trcdevid bit assignments

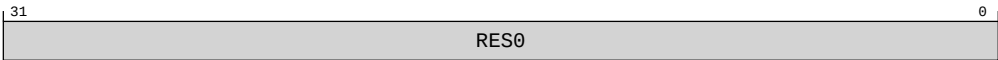


Table B-255: TRCDEVID bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFC8  | TRCDEVID | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.25 TRCDEVTYPE, Device Type Register

Provides discovery information for the component. If the part number field is not recognized, a debugger can report the information that is provided by TRCDEVTYPE about the component instead.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xFCC

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-122: ext\_trcdevtype bit assignments



Table B-257: TRCDEVTYPE bit descriptions

| Bits   | Name  | Description                                                                                     | Reset  |
|--------|-------|-------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                        | RES0   |
| [7:4]  | SUB   | Component sub-type.<br><b>0b0001</b><br>When MAJOR == 0x3 (Trace source): Associated with a PE. | 0b0001 |
| [3:0]  | MAJOR | Component major type.<br><b>0b0011</b><br>Trace source.                                         | 0b0011 |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| ETE       | 0xFCC  | TRCDEVTYPE | None  |

This interface is accessible as follows:

**When !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

B.7.26 TRCPIDR4, Peripheral Identification Register 4

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xFD0

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0100 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-123: ext\_trcpidr4 bit assignments



**Table B-259: TRCPIDR4 bit descriptions**

| Bits   | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset  |
|--------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RES0   |
| [7:4]  | SIZE  | <p>Size of the component.</p> <p>The distance from the start of the address space used by this component to the end of the component identification registers.</p> <p>A value of 0b0000 means one of the following is true:</p> <ul style="list-style-type: none"> <li>The component uses a single 4KB block.</li> <li>The component uses an <b>IMPLEMENTATION DEFINED</b> number of 4KB blocks.</li> </ul> <p>Any other value means the component occupies <math>2^{\text{TRCPIDR4.SIZE}}</math> 4KB blocks.</p> <p><b>0b0000</b></p> | 0b0000 |
| [3:0]  | DES_2 | <p>Designer, JEP106 continuation code. This is the JEDEC-assigned JEP106 bank identifier for the designer of the component, minus 1. The code identifies the designer of the component, which might not be the same as the implementer of the device containing the component. To obtain a number, or to see the assignment of these codes, contact JEDEC <a href="http://www.jedec.org">http://www.jedec.org</a>.</p> <p><b>0b0100</b></p> <p>Arm Limited. This is bits[3:0] of the JEP106 continuation code.</p>                     | 0b0100 |

## Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFD0  | TRCPIDR4 | None  |

This interface is accessible as follows:

### When !IsTraceCorePowered()

ERROR

### Otherwise

RO

## B.7.27 TRCPIDR5, Peripheral Identification Register 5

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

Component

ETE

Register offset

0xFD4

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-124: ext\_trcpidr5 bit assignments

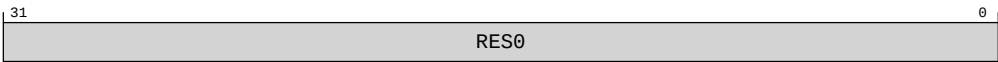


Table B-261: TRCPIDR5 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFD4  | TRCPIDR5 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO



### B.7.28 TRCPIDR6, Peripheral Identification Register 6

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

ETE

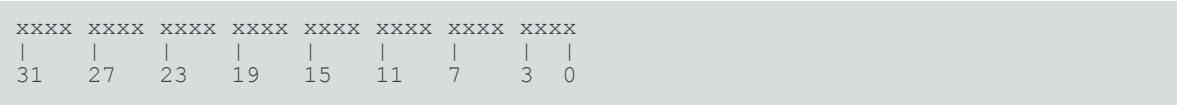
##### Register offset

0xFD8

##### Access type

See bit descriptions

##### Reset value



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-125: ext\_trcpidr6 bit assignments



Table B-263: TRCPIDR6 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

#### Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFD8  | TRCPIDR6 | None  |

This interface is accessible as follows:

**When !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

### B.7.29 TRCPIDR7, Peripheral Identification Register 7

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

#### Configurations

This register is available in all configurations.

#### Attributes

**Width**

32

**Component**

ETE

**Register offset**

0xFDC

**Access type**

See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-126: ext\_trcpidr7 bit assignments

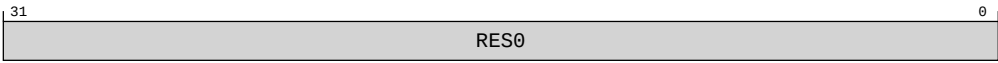


Table B-265: TRCPIDR7 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:0] | RES0 | Reserved    | RES0  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFDC  | TRCPIDR7 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.30 TRCPIDR0, Peripheral Identification Register 0

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

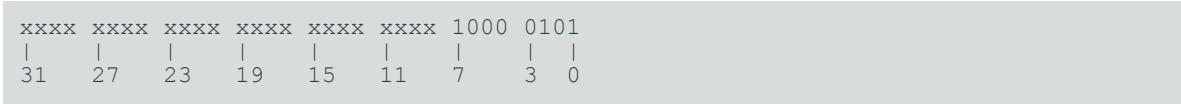
Register offset

0xFE0

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-127: ext\_trcpidr0 bit assignments



Table B-267: TRCPIDR0 bit descriptions

| Bits   | Name   | Description                                                                                                                                                                                                                                | Reset |
|--------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:8] | RES0   | Reserved                                                                                                                                                                                                                                   | RES0  |
| [7:0]  | PART_0 | Part number, bits [7:0].<br><br>The part number is selected by the designer of the component, and is stored in ext-TRCPIDR1.PART_1 and TRCPIDR0.PART_0.<br><br><b>0b10000101</b><br><br>Least significant byte of the ETM trace unit part. | 0x85  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFE0  | TRCPIDR0 | None  |

This interface is accessible as follows:

**When !IsTraceCorePowered()**

ERROR

**Otherwise**

RO

B.7.31 TRCPIDR1, Peripheral Identification Register 1

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

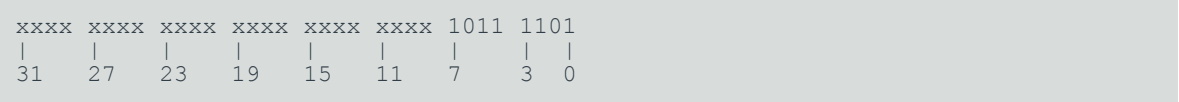
Register offset

0xFE4

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-128: ext\_trcpidr1 bit assignments

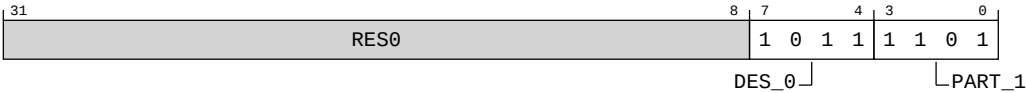


Table B-269: TRCPIDR1 bit descriptions

| Bits   | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset  |
|--------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0   |
| [7:4]  | DES_0 | Designer, JEP106 identification code, bits [3:0]. TRCPIDR1.DES_0 and ext-TRCPIDR2.DES_1 together form the JEDEC-assigned JEP106 identification code for the designer of the component. The parity bit in the JEP106 identification code is not included. The code identifies the designer of the component, which might not be not the same as the implementer of the device containing the component. To obtain a number, or to see the assignment of these codes, contact JEDEC <a href="http://www.jedec.org">http://www.jedec.org</a> .<br><br><b>0b1011</b><br>Arm Limited. This is the least significant nibble of JEP106 ID code. | 0b1011 |

| Bits  | Name   | Description                                                                                                                                                                                                            | Reset  |
|-------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [3:0] | PART_1 | Part number, bits [11:8].<br><br>The part number is selected by the designer of the component, and is stored in TRCPIDR1.PART_1 and ext-TRCPIDR0.PART_0.<br><br><b>0b1101</b><br>Part number, most significant nibble. | 0b1101 |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFE4  | TRCPIDR1 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.32 TRCPIDR2, Peripheral Identification Register 2

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xFE8

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 1011 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-129: ext\_trcpidr2 bit assignments

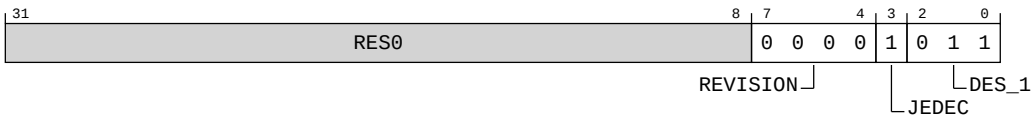


Table B-271: TRCPIDR2 bit descriptions

| Bits   | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset  |
|--------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:8] | RES0     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RES0   |
| [7:4]  | REVISION | Component major revision. TRCPIDR2.REVISION and ext-TRCPIDR3.REVAND together form the revision number of the component, with TRCPIDR2.REVISION being the most significant part and ext-TRCPIDR3.REVAND the least significant part. When a component is changed, TRCPIDR2.REVISION or ext-TRCPIDR3.REVAND are increased to ensure that software can differentiate the different revisions of the component. ext-TRCPIDR3.REVAND should be set to 0b0000 when TRCPIDR2.REVISION is increased.<br><br>0b0000<br>rOp2 - Part major revision.                                                                            | 0b0000 |
| [3]    | JEDEC    | JEDEC-assigned JEP106 implementer code is used.<br><br>0b1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0b1    |
| [2:0]  | DES_1    | Designer, JEP106 identification code, bits [6:4]. ext-TRCPIDR1.DES_0 and TRCPIDR2.DES_1 together form the JEDEC-assigned JEP106 identification code for the designer of the component. The parity bit in the JEP106 identification code is not included. The code identifies the designer of the component, which might not be not the same as the implementer of the device containing the component. To obtain a number, or to see the assignment of these codes, contact JEDEC <a href="http://www.jedec.org">http://www.jedec.org</a> .<br><br>0b011<br>Arm Limited. Most significant nibble of JEP106 ID code. | 0b011  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFE8  | TRCPIDR2 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

### B.7.33 TRCPIDR3, Peripheral Identification Register 3

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Component

ETE

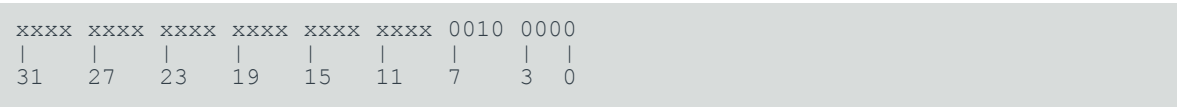
##### Register offset

0xFEC

##### Access type

See bit descriptions

##### Reset value



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-130: ext\_trcpidr3 bit assignments

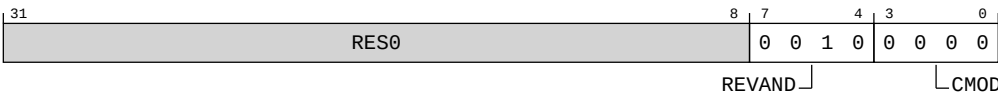


Table B-273: TRCPIDR3 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [31:8] | RES0 | Reserved    | RES0  |



| Bits  | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset  |
|-------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [7:4] | REVAND | <p>Component minor revision. ext-TRCPIDR2.REVISION and TRCPIDR3.REVAND together form the revision number of the component, with ext-TRCPIDR2.REVISION being the most significant part and TRCPIDR3.REVAND the least significant part. When a component is changed, ext-TRCPIDR2.REVISION or TRCPIDR3.REVAND are increased to ensure that software can differentiate the different revisions of the component. TRCPIDR3.REVAND should be set to 0b0000 when ext-TRCPIDR2.REVISION is increased.</p> <p><b>0b0010</b></p> <p>Part minor revision.</p> | 0b0010 |
| [3:0] | CMOD   | <p>Customer Modified.</p> <p>Indicates the component has been modified.</p> <p>A value of 0b0000 means the component is not modified from the original design.</p> <p>Any other value means the component has been modified in an <b>IMPLEMENTATION DEFINED</b> way.</p> <p><b>0b0000</b></p> <p>Not Customer modified.</p>                                                                                                                                                                                                                         | 0b0000 |

### Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFEC  | TRCPIDR3 | None  |

This interface is accessible as follows:

#### When !IsTraceCorePowered()

ERROR

#### Otherwise

RO

## B.7.34 TRCCIDR0, Component Identification Register 0

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Component

ETE

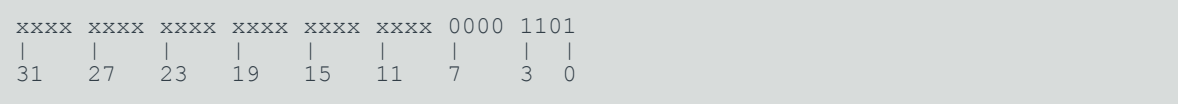
Register offset

0xFF0

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-131: ext\_trccidr0 bit assignments

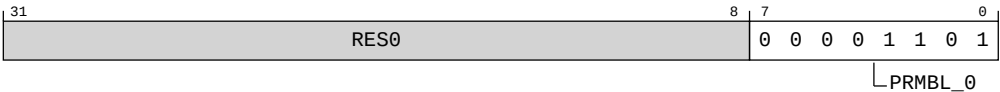


Table B-275: TRCCIDR0 bit descriptions

| Bits   | Name    | Description                                                     | Reset |
|--------|---------|-----------------------------------------------------------------|-------|
| [31:8] | RES0    | Reserved                                                        | RES0  |
| [7:0]  | PRMBL_0 | Component identification preamble, segment 0.<br><br>0b00001101 | 0x0D  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFF0  | TRCCIDR0 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.35 TRCCIDR1, Component Identification Register 1

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xFF4

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-132: ext\_trccidr1 bit assignments



Table B-277: TRCCIDR1 bit descriptions

| Bits   | Name  | Description                                                | Reset  |
|--------|-------|------------------------------------------------------------|--------|
| [31:8] | RES0  | Reserved                                                   | RES0   |
| [7:4]  | CLASS | Component class.<br><b>0b1001</b><br>CoreSight peripheral. | 0b1001 |

| Bits  | Name    | Description                                                 | Reset  |
|-------|---------|-------------------------------------------------------------|--------|
| [3:0] | PRMBL_1 | Component identification preamble, segment 1.<br><br>0b0000 | 0b0000 |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFF4  | TRCCIDR1 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

B.7.36 TRCCIDR2, Component Identification Register 2

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

Register offset

0xFF8

Access type

See bit descriptions

Reset value

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0101 |
|      |      |      |      |      |      |      |      |
| 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-133: ext\_trccidr2 bit assignments

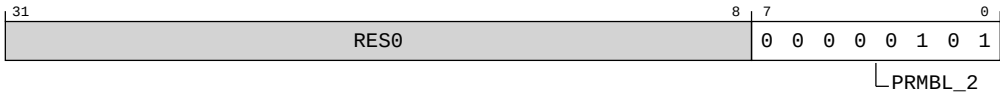


Table B-279: TRCCIDR2 bit descriptions

| Bits   | Name    | Description                                                 | Reset |
|--------|---------|-------------------------------------------------------------|-------|
| [31:8] | RES0    | Reserved                                                    | RES0  |
| [7:0]  | PRMBL_2 | Component identification preamble, segment 2.<br>0b00000101 | 0x05  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFF8  | TRCCIDR2 | None  |

This interface is accessible as follows:

When `!IsTraceCorePowered()`

ERROR

Otherwise

RO

B.7.37 TRCCIDR3, Component Identification Register 3

Provides discovery information about the component.

For additional information, see the CoreSight Architecture Specification.

Configurations

This register is available in all configurations.

Attributes

Width

32

Component

ETE

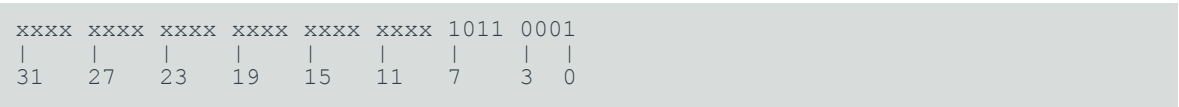
Register offset

0xFFC

Access type

See bit descriptions

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-134: ext\_trccidr3 bit assignments



Table B-281: TRCCIDR3 bit descriptions

| Bits   | Name    | Description                                                        | Reset |
|--------|---------|--------------------------------------------------------------------|-------|
| [31:8] | RES0    | Reserved                                                           | RES0  |
| [7:0]  | PRMBL_3 | Component identification preamble, segment 3.<br><b>0b10110001</b> | 0xB1  |

Accessibility

External debugger accesses to this register are unaffected by the OS Lock.

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| ETE       | 0xFFC  | TRCCIDR3 | None  |

This interface is accessible as follows:

When !IsTraceCorePowered()

ERROR

Otherwise

RO

## B.8 External RAS registers summary

The following summary table provides an overview of all memory-mapped RAS registers in the core.

For more information on registers listed in the table, click on the link associated with the register name.

For registers without a listed reset value, refer to the individual field resets documented on the register description pages or to the [Arm® Architecture Reference Manual for A-profile architecture](#).

**Table B-283: RAS registers summary**

| Offset | Name                       | Reset                      | Width  | Description                                                 |
|--------|----------------------------|----------------------------|--------|-------------------------------------------------------------|
| 0x0    | <a href="#">ERROFR</a>     | See individual bit resets. | 64-bit | Error Record <n> Feature Register                           |
| 0x8    | <a href="#">ERROCTLR</a>   | See individual bit resets. | 64-bit | Error Record <n> Control Register                           |
| 0x10   | <a href="#">ERROSTATUS</a> | See individual bit resets. | 64-bit | Error Record <n> Primary Status Register                    |
| 0x18   | <a href="#">ERROADDR</a>   | See individual bit resets. | 64-bit | Error Record <n> Address Register                           |
| 0x20   | <a href="#">ERROMISC0</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 0                   |
| 0x28   | <a href="#">ERROMISC1</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 1                   |
| 0x30   | <a href="#">ERROMISC2</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 2                   |
| 0x38   | <a href="#">ERROMISC3</a>  | See individual bit resets. | 64-bit | Error Record <n> Miscellaneous Register 3                   |
| 0x800  | <a href="#">ERROPFGF</a>   | See individual bit resets. | 64-bit | Error Record <n> Pseudo-fault Generation Feature Register   |
| 0x808  | <a href="#">ERROPFGCTL</a> | See individual bit resets. | 64-bit | Error Record <n> Pseudo-fault Generation Control Register   |
| 0x810  | <a href="#">ERROPFGCDN</a> | See individual bit resets. | 64-bit | Error Record <n> Pseudo-fault Generation Countdown Register |

### B.8.1 ERROFR, Error Record <n> Feature Register

Defines whether error record <n> is the first record owned by a node:

- If error record <n> is the first error record owned by a node, then ERR<n>FR.ED is not 0b00.
- If error record <n> is not the first error record owned by a node, then ERR<n>FR.ED is 0b00.

If error record <n> is the first record owned by the node, defines which of the common architecturally-defined features are implemented by the node and, of the implemented features, which are software programmable.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

**Component**

RAS

**Register offset**

0x0

**Access type**

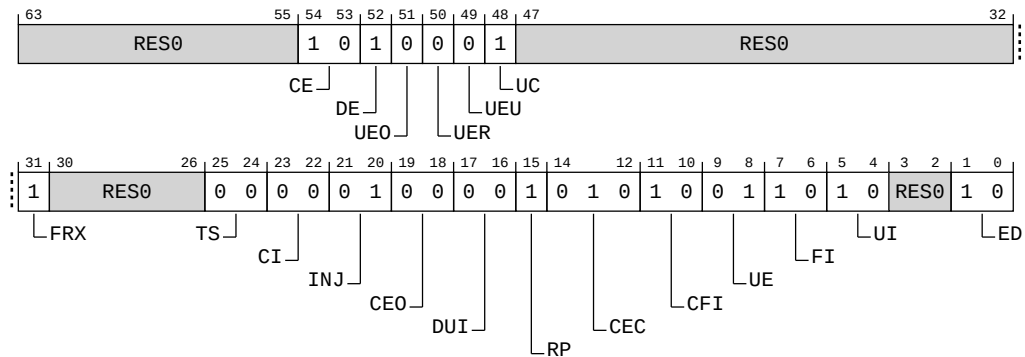
RO

**Reset value**

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | x101 | 0001 | xxxx | xxxx | xxxx | xxxx | 1xxx | xx00 | 0001 | 0000 | 1010 | 1001 | 1010 | xx10 |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0    |

**Note**

Where the reset reads xxxx, see individual bits.

**Bit descriptions****Figure B-135: ext\_err0fr bit assignments****Table B-284: ERR0FR bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                           | Reset |
|---------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:55] | RES0 | Reserved                                                                                                                                                                                                                              | RES0  |
| [54:53] | CE   | Corrected Error recording. Describes the types of Corrected errors the node can record, if any.<br><b>0b10</b><br>Records only non-specific Corrected errors. That is, Corrected errors recorded by setting ERXSTATUS_EL1.CE to 0b10. | 0b10  |
| [52]    | DE   | Deferred Error recording. Describes whether the node supports recording Deferred errors.<br><b>0b1</b><br>Records Deferred errors.                                                                                                    | 0b1   |



| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [51]    | UEO  | Latent or Restartable Error recording. Describes whether the node supports recording Latent or Restartable errors.<br><b>0b0</b><br>Does not record Latent or Restartable errors.                                                                                                                                                                                                            | 0b0   |
| [50]    | UER  | Signaled or Recoverable Error recording. Describes whether the node supports recording Signaled or Recoverable errors.<br><b>0b0</b><br>Does not record Signaled or Recoverable errors.                                                                                                                                                                                                      | 0b0   |
| [49]    | UEU  | Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.<br><b>0b0</b><br>Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.                                                                                                                                                                       | 0b0   |
| [48]    | UC   | Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.<br><b>0b1</b><br>Uncontainable Error recording. Describes whether the node supports recording Uncontainable errors.                                                                                                                                                                       | 0b1   |
| [47:32] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                     | RES0  |
| [31]    | FRX  | Feature Register extension. Defines whether ERXFR_EL1[63:48] are architecturally defined.<br><b>0b1</b><br>ERXFR_EL1[63:48] are defined by the architecture.                                                                                                                                                                                                                                 | 0b1   |
| [30:26] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                     | RES0  |
| [25:24] | TS   | Timestamp Extension. Indicates whether, for each error record <m> owned by this node, ERXMISC3_EL1 is used as the timestamp register, and, if it is, the timebase used by the timestamp.<br><b>0b00</b><br>The node does not support a timestamp register.                                                                                                                                   | 0b00  |
| [23:22] | CI   | Critical error interrupt. Indicates whether the critical error interrupt and associated controls are implemented.<br><b>0b00</b><br>Does not support the critical error interrupt. ERXCTLR_EL1.CI is RES0.                                                                                                                                                                                   | 0b00  |
| [21:20] | INJ  | Fault Injection Extension. Indicates whether the RAS Common Fault Injection Model Extension is implemented.<br><b>0b01</b><br>The node implements the RAS Common Fault Injection Model Extension. See ERXPFGF_EL1 for more information.                                                                                                                                                      | 0b01  |
| [19:18] | CEO  | Corrected Error overwrite. Indicates the behavior when a second Corrected error is detected after a first Corrected error has been recorded by an error record <m> owned by the node.<br><b>0b00</b><br>Counts Corrected errors if a counter is implemented. Keeps the previous error syndrome. If the counter overflows, or no counter is implemented, then ERXSTATUS_EL1.OF is set to 0b1. | 0b00  |
| [17:16] | DUI  | Error recovery interrupt for deferred errors control. Indicates whether the control for enabling error recovery interrupts on deferred errors are implemented.<br><b>0b00</b><br>Does not support the control for enabling error recovery interrupts on deferred errors. ERXCTLR_EL1.DUI is RES0.                                                                                            | 0b00  |

| Bits    | Name        | Description                                                                                                                                                                                                                                                                                                                                                                  | Reset       |
|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| [15]    | RP          | Repeat counter. Indicates whether the node implements the repeat Corrected error counter in ERXMISCO_EL1 for each error record <m> owned by the node that implements the standard Corrected error counter.<br><br><b>0b1</b><br><br>A first (repeat) counter and a second (other) counter are implemented. The repeat counter is the same size as the primary error counter. | 0b1         |
| [14:12] | CEC         | Corrected Error Counter. Indicates whether the node implements the standard Corrected error counter (CE counter) mechanisms in ERXMISCO_EL1 for each error record <m> owned by the node that can record countable errors.<br><br><b>0b010</b><br><br>Implements an 8-bit Corrected error counter in ERXMISCO_EL1[39:32].                                                     | 0b010       |
| [11:10] | CFI         | Fault handling interrupt for corrected errors. Indicates whether the control for enabling fault handling interrupts on corrected errors are implemented.<br><br><b>0b10</b><br><br>Control for enabling fault handling interrupts on corrected errors is supported and controllable using ERXCTLR_EL1.CFI.                                                                   | 0b10        |
| [9:8]   | UE          | In-band uncorrected error reporting. Indicates whether the in-band uncorrected error reporting (External Aborts) and associated controls are implemented.<br><br><b>0b01</b><br><br>In-band uncorrected error reporting (External Aborts) is supported and always enabled. ERXCTLR_EL1.UE is <b>RES0</b> .                                                                   | 0b01        |
| [7:6]   | FI          | Fault handling interrupt. Indicates whether the fault handling interrupt and associated controls are implemented.<br><br><b>0b10</b><br><br>Fault handling interrupt is supported and controllable using ERXCTLR_EL1.FI.                                                                                                                                                     | 0b10        |
| [5:4]   | UI          | Error recovery interrupt for uncorrected errors. Indicates whether the error handling interrupt and associated controls are implemented.<br><br><b>0b10</b><br><br>Error handling interrupt is supported and controllable using ERXCTLR_EL1.UI.                                                                                                                              | 0b10        |
| [3:2]   | <b>RES0</b> | Reserved                                                                                                                                                                                                                                                                                                                                                                     | <b>RES0</b> |
| [1:0]   | ED          | Error reporting and logging. Indicates whether error record <n> is the first record owned the node, and, if so, whether it implements the controls for enabling and disabling error reporting and logging.<br><br><b>0b10</b><br><br>Error reporting and logging is controllable using ERXCTLR_EL1.ED.                                                                       | 0b10        |

## Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| RAS       | 0x0    | ERROFR   | None  |

This interface is accessible as follows:

RO

B.8.2 ERR0CTLR, Error Record <n> Control Register

The error control register contains enable bits for the node that writes to this record:

- Enabling error detection and correction.
- Enabling the critical error, error recovery, and fault handling interrupts.
- Enabling in-band error response for uncorrected errors.

For each bit, if the node does not support the feature, then the bit is **RES0**. The definition of each record is IMPLEMENTATION DEFINED.

Configurations

ext-ERR<n>FR contains additional information about the node.

Attributes

Width

64

Component

RAS

Register offset

0x8

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxx0 | xxxx | 00x0 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-136: ext\_err0ctlr bit assignments

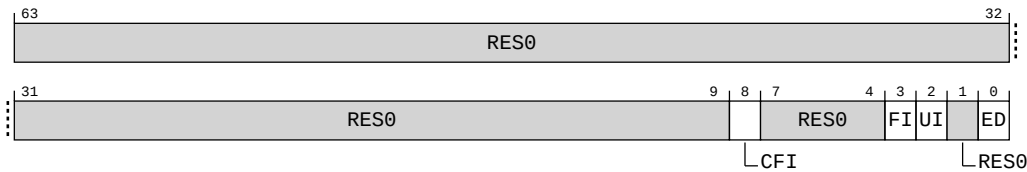


**Table B-286: ERROCTLR bit descriptions**

| Bits   | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|--------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:9] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RES0  |
| [8]    | CFI  | <p>Fault handling interrupt for Corrected errors enable.</p> <p>This control applies to errors arising from both reads and writes.</p> <p>The fault handling interrupt is generated when one of the standard CE counters on ERXMISC0_EL1 overflows and the overflow bit is set. The possible values are:</p> <p><b>0b0</b></p> <p>Fault handling interrupt not generated for Corrected errors.</p> <p><b>0b1</b></p> <p>Fault handling interrupt generated for Corrected errors.</p> <p>The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.</p> <p>Cold reset only. Unaffected by Warm reset</p> | 0b0   |
| [7:4]  | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RES0  |
| [3]    | FI   | <p>Fault handling interrupt enable.</p> <p>This control applies to errors arising from both reads and writes.</p> <p>The fault handling interrupt is generated for all detected Deferred errors and Uncorrected errors. The possible values are:</p> <p><b>0b0</b></p> <p>Fault handling interrupt disabled.</p> <p><b>0b1</b></p> <p>Fault handling interrupt enabled.</p> <p>The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                          | 0b0   |
| [2]    | UI   | <p>Uncorrected error recovery interrupt enable.</p> <p>This control applies to errors arising from both reads and writes.</p> <p>When enabled, the error recovery interrupt is generated for all detected Uncorrected errors that are not deferred.</p> <p><b>0b0</b></p> <p>Error recovery interrupt disabled.</p> <p><b>0b1</b></p> <p>Error recovery interrupt enabled.</p> <p>The interrupt is generated even if the error syndrome is discarded because the error record already records a higher priority error.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                       | 0b0   |
| [1]    | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RES0  |

| Bits | Name | Description                                                                                                                                                                                                                                 | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | ED   | Error Detection and correction enable. The possible values are:<br><br><b>0b0</b><br>Error detection and correction disabled.<br><br><b>0b1</b><br>Error detection and correction enabled.<br><br>Cold reset only. Unaffected by Warm reset | 0b0   |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| RAS       | 0x8    | ERROCTL  | None  |

This interface is accessible as follows:

RW

### B.8.3 ERROSTATUS, Error Record <n> Primary Status Register

Contains status information for error record <n>, including:

- Whether any error has been detected (valid).
- Whether any detected error was not corrected, and returned to a Requester.
- Whether any detected error was not corrected and deferred.
- Whether an error record has been discarded because additional errors have been detected before the first error was handled by software (overflow).
- Whether any error has been reported.
- Whether the other error record registers contain valid information.
- Whether the error was reported because poison data was detected or because a corrupt value was detected by an error detection code.
- A primary error code.
- An **IMPLEMENTATION DEFINED** extended error code.

Within this register:

- ERR<n>STATUS.{AV, V, MV} are valid bits that define whether error record <n> registers are valid.
- ERR<n>STATUS.{UE, OF, CE, DE, UET} encode the types of error or errors recorded.
- ERR<n>STATUS.{CI, ER, PN, IERR, SERR} are syndrome fields.

### Configurations

ext-ERR<n>FR describes the features implemented by the node.

Attributes

Width

64

Component

RAS

Register offset

0x10

Access type

Read

R

Write

W

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 0000 | 0000 | 0000 | xxxx | xxxx | xxxx | xxx0 | 0000 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3    |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 0    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-137: ext\_err0status bit assignments

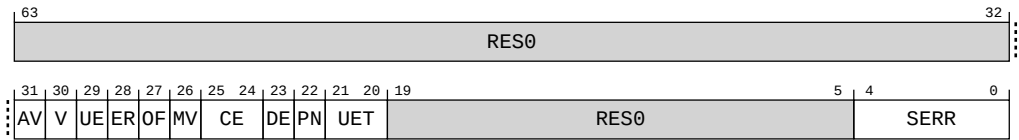


Table B-288: ERR0STATUS bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:32] | RES0 | Reserved    | RES0  |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31] | AV   | <p>Address Valid. The possible values are:</p> <p><b>0b0</b></p> <p>ERXADDR_EL1 not valid.</p> <p><b>0b1</b></p> <p>ERXADDR_EL1 contains an address associated with the highest priority error recorded by this record.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                                                                                                                                                | 0b0   |
| [30] | V    | <p>Status Register Valid. The possible values are:</p> <p><b>0b0</b></p> <p>ERXSTATUS_EL1 not valid.</p> <p><b>0b1</b></p> <p>ERXSTATUS_EL1 valid. At least one error has been recorded.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                                                                                                                                                                               | 0b0   |
| [29] | UE   | <p>Uncorrected Error. The possible values are:</p> <p><b>0b0</b></p> <p>No errors have been detected, or all detected errors have been either corrected or deferred.</p> <p><b>0b1</b></p> <p>At least one detected error was not corrected and not deferred.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p> | 0b0   |
| [28] | ER   | <p>Error Reported. The possible values are:</p> <p><b>0b0</b></p> <p>No in-band error (External Abort) reported.</p> <p><b>0b1</b></p> <p>An External Abort was signaled by the node to the master making the access or other transaction.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p> <p><b>Note:</b></p> <p>An External Abort signaled by the node might be masked and not generate any exception.</p>                                                                                                                                           | 0b0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [27] | OF   | <p>Overflow. The possible values are:</p> <p><b>0b0</b></p> <p>If UE == 1, then no error status for an Uncorrected error has been discarded.</p> <p>If UE == 0 and DE == 1, then no error status for a Deferred error has been discarded.</p> <p>If UE == 0, DE == 0, and CE != 0b00, then the corrected error counter has not overflowed.</p> <p><b>0b1</b></p> <p>More than one error has occurred and so details of the other error have been discarded.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>Cold reset only. Unaffected by Warm reset</p> <p>This bit is read/write-one-to-clear.</p> | 0b0   |
| [26] | MV   | <p>Miscellaneous Registers Valid. The possible values are:</p> <p><b>0b0</b></p> <p>ERXMISC&lt;m&gt;_EL1 not valid.</p> <p><b>0b1</b></p> <p>This bit indicates that the ERXMISC&lt;m&gt;_EL1 registers contain additional information for an error recorded by this record.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p> <p><b>Note:</b></p> <p>If the ERXMISC&lt;m&gt;_EL1 registers can contain additional information for a previously recorded error, then the contents must be self-describing to software or a user. For example, certain fields might relate only to Corrected errors, and other fields only to the most recent error that was not discarded.</p>                                                                         | 0b0   |



| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|---------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [25:24] | CE   | <p>Corrected Error. The possible values are:</p> <p><b>0b00</b><br/>No errors were corrected.</p> <p><b>0b01</b><br/>At least one transient error was corrected.</p> <p><b>0b10</b><br/>At least one error was corrected.</p> <p><b>0b11</b><br/>At least one persistent error was corrected.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this field is nonzero, then Arm recommends that software write ones to this field to clear this field to zero.</p> <p>This field is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>This field is read/write-ones-to-clear. Writing a value other than all-zeros or all-ones sets this field to an <b>UNKNOWN</b> value.</p> <p>Cold reset only. Unaffected by Warm reset</p> | 0b00  |
| [23]    | DE   | <p>Deferred Error. The possible values are:</p> <p><b>0b0</b><br/>No errors were deferred.</p> <p><b>0b1</b><br/>At least one error was not corrected and deferred.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if ERXSTATUS_EL1.V == 0b0.</p> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                                                                                                    | 0b0   |
| [22]    | PN   | <p>Poison. The value is:</p> <p><b>0b0</b><br/>This core cannot distinguish a poisoned value from a corrupted value.</p> <p>When clearing ERXSTATUS_EL1.V to 0b0, if this bit is nonzero, then Arm recommends that software write 0b1 to this bit to clear this bit to zero.</p> <p>This bit is not valid and reads <b>UNKNOWN</b> if any of the following are true:</p> <ul style="list-style-type: none"> <li>ERXSTATUS_EL1.V == 0b0.</li> <li>ERXSTATUS_EL1.{DE,UE} == {0,0}.</li> </ul> <p>This bit is read/write-one-to-clear.</p> <p>Cold reset only. Unaffected by Warm reset</p>                                                                                                                                                         | 0b0   |

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset    |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| [21:20] | UET  | Uncorrected Error Type. The value is:<br><br><b>0b00</b><br>Uncorrected error, Uncontainable error (UC).<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0b00     |
| [19:5]  | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RES0     |
| [4:0]   | SERR | Primary error code.<br><br>The primary error code might be used by a fault handling agent to triage an error without requiring device-specific code. For example, to count and threshold corrected errors in software, or generate a short log entry.<br><br>The possible values are:<br><br><b>0b000000</b><br>No error<br><br><b>0b000010</b><br>ECC error from internal data buffer.<br><br><b>0b000110</b><br>ECC error on cache data RAM.<br><br><b>0b000111</b><br>ECC error on cache tag or dirty RAM.<br><br><b>0b010000</b><br>Parity error on TLB data RAM.<br><br><b>0b100010</b><br>Error response for a cache copyback.<br><br><b>0b101010</b><br>Deferred error from slave not supported at the consumer. For example, poisoned data received from a slave by a master that cannot defer the error further.<br><br>Cold reset only. Unaffected by Warm reset | 0b000000 |

## Access

ERR<n>STATUS.{AV, V, UE, ER, OF, MV, CE, DE, PN, UET, CI} are write-one-to-clear (W1C) fields, meaning writes of zero are ignored, and a write of one or all-ones to the field clears the field to zero. ERR<n>STATUS.{IERR, SERR} are read/write (RW) fields, although the set of implemented valid values is **IMPLEMENTATION DEFINED**. See also ext-ERR<n>PFGF.SYN.

After reading ERR<n>STATUS, software must clear the valid fields in the register to allow new errors to be recorded. However, between reading the register and clearing the valid fields, a new error might have overwritten the register. To prevent this error being lost by software, the register prevents updates to fields that might have been updated by a new error.

When RAS System Architecture v1.0 is implemented:

- Writes to ERR<n>STATUS.{UE, DE, CE} are ignored if ERR<n>STATUS.OF is 1 and is not being cleared to 0.

- Writes to ERR<n>STATUS.V are ignored if any of ERR<n>STATUS.{UE, DE, CE} are nonzero and are not being cleared to zero.
- Writes to ERR<n>STATUS.{AV, MV} and the ERR<n>STATUS.{ER, PN, UET, IERR, SERR} syndrome fields are ignored if the highest priority nonzero error status field is not being cleared to zero. The error status fields in priority order from highest to lowest, are ERR<n>STATUS.UE, ERR<n>STATUS.DE, and ERR<n>STATUS.CE.

When RAS System Architecture v1.1 is implemented, a write to the register is ignored if all of:

- Any of ERR<n>STATUS.{V, UE, OF, CE, DE} are nonzero before the write.
- The write does not clear the nonzero ERR<n>STATUS.{V, UE, OF, CE, DE} fields to zero by writing ones to the applicable field or fields.

Some of the fields in ERR<n>STATUS are also defined as **UNKNOWN** where certain combinations of ERR<n>STATUS.{V, DE, UE} are zero. The rules for writes to ERR<n>STATUS allow a node to implement such a field as a fixed read-only value.

For example, when RAS System Architecture v1.1 is implemented, a write to ERR<n>STATUS when ERR<n>STATUS.V is 1 results in either ERR<n>STATUS.V field being cleared to zero, or ERR<n>STATUS.V not changing. Since all fields in ERR<n>STATUS, other than ERR<n>STATUS.{AV, V, MV}, usually read as **UNKNOWN** values when ERR<n>STATUS.V is zero, this means those fields can be implemented as read-only if applicable.

To ensure correct and portable operation, when software is clearing the valid fields in the register to allow new errors to be recorded, Arm recommends that software performs the following sequence of operations in order:

1. Read ERR<n>STATUS and determine which fields need to be cleared to zero.
2. In a single write to ERR<n>STATUS:
  - Write ones to all the W1C fields that are nonzero in the read value.
  - Write zero to all the W1C fields that are zero in the read value.
  - Write zero to all the RW fields.
3. Read back ERR<n>STATUS after the write to confirm no new fault has been recorded.

Otherwise, these fields might not have the correct value when a new fault is recorded.

## Accessibility

ERR<n>STATUS.{AV, V, UE, ER, OF, MV, CE, DE, PN, UET, CI} are write-one-to-clear (W1C) fields, meaning writes of zero are ignored, and a write of one or all-ones to the field clears the field to zero. ERR<n>STATUS.{IERR, SERR} are read/write (RW) fields, although the set of implemented valid values is IMPLEMENTATION DEFINED. See also ext-ERR<n>PFGF.SYN.

After reading ERR<n>STATUS, software must clear the valid fields in the register to allow new errors to be recorded. However, between reading the register and clearing the valid fields, a new error might have overwritten the register. To prevent this error being lost by software, the register prevents updates to fields that might have been updated by a new error.

When RAS System Architecture v1.0 is implemented:

- Writes to ERR<n>STATUS.{UE, DE, CE} are ignored if ERR<n>STATUS.OF is 1 and is not being cleared to 0.
- Writes to ERR<n>STATUS.V are ignored if any of ERR<n>STATUS.{UE, DE, CE} are nonzero and are not being cleared to zero.
- Writes to ERR<n>STATUS.{AV, MV} and the ERR<n>STATUS.{ER, PN, UET, IERR, SERR} syndrome fields are ignored if the highest priority nonzero error status field is not being cleared to zero. The error status fields in priority order from highest to lowest, are ERR<n>STATUS.UE, ERR<n>STATUS.DE, and ERR<n>STATUS.CE.

When RAS System Architecture v1.1 is implemented, a write to the register is ignored if all of:

- Any of ERR<n>STATUS.{V, UE, OF, CE, DE} are nonzero before the write.
- The write does not clear the nonzero ERR<n>STATUS.{V, UE, OF, CE, DE} fields to zero by writing ones to the applicable field or fields.

Some of the fields in ERR<n>STATUS are also defined as UNKNOWN where certain combinations of ERR<n>STATUS.{V, DE, UE} are zero. The rules for writes to ERR<n>STATUS allow a node to implement such a field as a fixed read-only value.

For example, when RAS System Architecture v1.1 is implemented, a write to ERR<n>STATUS when ERR<n>STATUS.V is 1 results in either ERR<n>STATUS.V field being cleared to zero, or ERR<n>STATUS.V not changing. Since all fields in ERR<n>STATUS, other than ERR<n>STATUS.{AV, V, MV}, usually read as UNKNOWN values when ERR<n>STATUS.V is zero, this means those fields can be implemented as read-only if applicable.

To ensure correct and portable operation, when software is clearing the valid fields in the register to allow new errors to be recorded, Arm recommends that software performs the following sequence of operations in order:

1. Read ERR<n>STATUS and determine which fields need to be cleared to zero.
2. In a single write to ERR<n>STATUS:
  - Write ones to all the W1C fields that are nonzero in the read value.
  - Write zero to all the W1C fields that are zero in the read value.
  - Write zero to all the RW fields.
3. Read back ERR<n>STATUS after the write to confirm no new fault has been recorded.

Otherwise, these fields might not have the correct value when a new fault is recorded.

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| RAS       | 0x10   | ERROSTATUS | None  |

This interface is accessible as follows:

RW

### B.8.4 ERR0ADDR, Error Record <n> Address Register

If an address is associated with a detected error, then it is written to ERR<n>ADDR when the error is recorded. It is **IMPLEMENTATION DEFINED** how the recorded address maps to the software-visible physical address. Software might have to reconstruct the actual physical addresses using the identity of the node and knowledge of the system.

#### Configurations

ERRFR[FirstRecordOfNode(n)] describes the features implemented by the node that owns error record <n>. FirstRecordOfNode(n) is the index of the first error record owned by the same node as error record <n>. If the node owns a single record then FirstRecordOfNode(n) = n.

#### Attributes

**Width**

64

**Component**

RAS

**Register offset**

0x18

**Access type**

See bit descriptions

**Reset value**

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-138: ext\_err0addr bit assignments

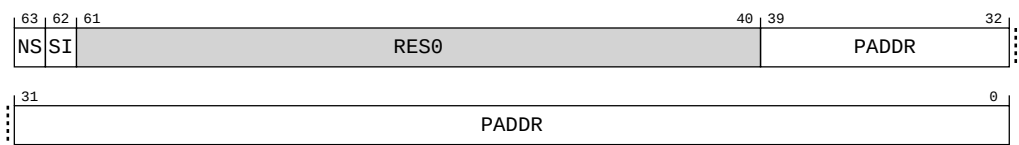


**Table B-290: ERR0ADDR bit descriptions**

| Bits    | Name  | Description                                                                                                                                                                                                                                                                                                                                                           | Reset  |
|---------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63]    | NS    | Non-secure attribute.<br><br><b>0b0</b><br>ERR<n>ADDR.PADDR is a Secure address.<br><br><b>0b1</b><br>ERR<n>ADDR.PADDR is a Non-secure address.                                                                                                                                                                                                                       | x      |
| [62]    | SI    | Secure Incorrect. Indicates whether ERR<n>ADDR.NS is valid.<br><br><b>0b0</b><br>ERR<n>ADDR.NS is correct. That is, it matches the programmers' view of the Non-secure attribute for the recorded location.<br><br><b>0b1</b><br>ERR<n>ADDR.NS might not be correct, and might not match the programmers' view of the Non-secure attribute for the recorded location. | x      |
| [61:40] | RES0  | Reserved                                                                                                                                                                                                                                                                                                                                                              | RES0   |
| [39:0]  | PADDR | Physical Address [39:0]. Address of the recorded location                                                                                                                                                                                                                                                                                                             | 40 {x} |

### Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| RAS       | 0x18   | ERR0ADDR | None  |

This interface is accessible as follows:

**When** `ERRPFGF[FirstRecordOfNode(0)].AV == '0' && ext-ERR0STATUS.AV == '1'`

RO

**Otherwise**

RW

## B.8.5 ERR0MISC0, Error Record <n> Miscellaneous Register 0

**IMPLEMENTATION DEFINED** error syndrome register. The miscellaneous syndrome registers might contain:

- Information to locate where the error was detected.
- If the error was detected within a FRU, the identity of the FRU.
- A Corrected error counter or counters.
- Other state information not present in the corresponding status and address registers.

If the node that owns error record <n> implements a standard format Corrected error counter or counters (`ERRFR[FirstRecordOfNode(n)].CEC != 0b000`), then it is **IMPLEMENTATION DEFINED** whether error record <n> can record countable errors, and:

- If error record <n> records countable errors, then `ERR<n>MISC0` implements the standard format Corrected error counter or counters for error record <n>.

- If error record <n> does not record countable errors, then it is recommended that the fields in ERR<n>MISC0 defined for the standard format counter or counters are **RES0**. That is, the fields behave like counters that never count.

Configurations

ERRFR[FirstRecordOfNode(n)] describes the features implemented by the node that owns error record <n>. FirstRecordOfNode(n) is the index of the first error record owned by the same node as error record <n>. If the node owns a single record then FirstRecordOfNode(n) = n.

For IMPLEMENTATION DEFINED fields in ERR<n>MISC0, writing zero returns the error record to an initial quiescent state.

In particular, if any IMPLEMENTATION DEFINED syndrome fields might generate a Fault Handling or Error Recovery Interrupt request, writing zero is sufficient to deactivate the Interrupt request.

Fields that are read-only, nonzero, and ignore writes are compliant with this requirement.



Arm recommends that any IMPLEMENTATION DEFINED syndrome field that can generate a Fault Handling, Error Recovery, Critical, or IMPLEMENTATION DEFINED, interrupt request is disabled at Cold reset and is enabled by software writing an IMPLEMENTATION DEFINED nonzero value to an IMPLEMENTATION DEFINED field in ERRCTRL[FirstRecordOfNode(n)].

Attributes

Width

64

Component

RAS

Register offset

0x20

Access type

Read

R

Write

W

Reset value

When ERRFR[FirstRecordOfNode(0)].CEC == '000'

XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

When ERRFR[FirstRecordOfNode(0)].CEC == '100' && ERRFR[FirstRecordOfNode(0)].RP == '0'

XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

When `ERRFR[FirstRecordOfNode(0)].CEC == '010' && ERRFR[FirstRecordOfNode(0)].RP == '0'`

XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

When `ERRFR[FirstRecordOfNode(0)].CEC == '100' && ERRFR[FirstRecordOfNode(0)].RP == '1'`

XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

When `ERRFR[FirstRecordOfNode(0)].CEC == '010' && ERRFR[FirstRecordOfNode(0)].RP == '1'`

XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX



Where the reset reads xxxx, see individual bits.

## Bit descriptions

When `ERRFR[FirstRecordOfNode(0)].CEC == '000'`

**Figure B-139: ext\_err0misc0 bit assignments**

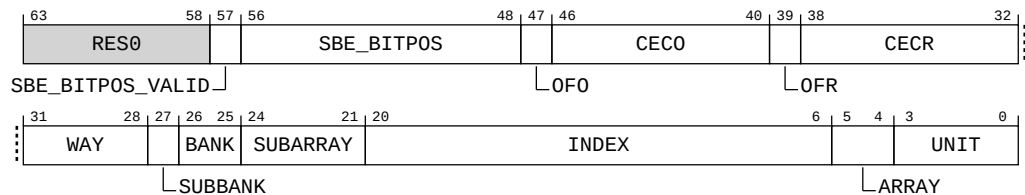


**Table B-292: ERR0MISC0 bit descriptions**

| Bits    | Name             | Description                                                                                                                                                                                                                       | Reset   |
|---------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| [63:58] | RES0             | Reserved                                                                                                                                                                                                                          | RES0    |
| [57]    | SBE_BITPOS_VALID | Single Bit Error (SBE) bit position field ERXMISC0_EL1.SBE_BITPOS contains valid data<br><br><b>0b0</b><br>ERXMISC0_EL1.SBE_BITPOS does not contain valid data.<br><br><b>0b1</b><br>ERXMISC0_EL1.SBE_BITPOS contains valid data. | x       |
| [56:48] | SBE_BITPOS       | Single Bit Error (SBE) bit position. For a correctable error in a RAM with ECC (L1 data cache, L2 cache), indicates the bit position of the corrected error. Valid when ERXMISC0_EL1.SBE_BITPOS_VALID is 1'b1                     | 9 { x } |



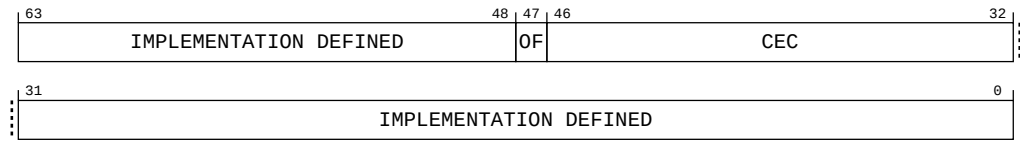
| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|---------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [47]    | OFO  | <p>Sticky overflow bit, other. Set to 1 when ERXMISCO_EL1.CECO is incremented and wraps through zero.</p> <p><b>0b0</b><br/>Other counter has not overflowed.</p> <p><b>0b1</b><br/>Other counter has overflowed.</p> <p>A direct write that modifies this bit might indirectly set ERXSTATUS_EL1.OF to an <b>UNKNOWN</b> value and a direct write to ERXSTATUS_EL1.OF that clears it to zero might indirectly set this bit to an <b>UNKNOWN</b> value.</p> <p>Unaffected by Cold or Warm reset.</p>    | x     |
| [46:40] | CECO | <p>Corrected error count, other. Incremented for each countable error that is not accounted for by incrementing ERXMISCO_EL1.CECR.</p> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                                                                                         | 7 {x} |
| [39]    | OFR  | <p>Sticky overflow bit, repeat. Set to 1 when ERXMISCO_EL1.CECR is incremented and wraps through zero.</p> <p><b>0b0</b><br/>Repeat counter has not overflowed.</p> <p><b>0b1</b><br/>Repeat counter has overflowed.</p> <p>A direct write that modifies this bit might indirectly set ERXSTATUS_EL1.OF to an <b>UNKNOWN</b> value and a direct write to ERXSTATUS_EL1.OF that clears it to zero might indirectly set this bit to an <b>UNKNOWN</b> value.</p> <p>Unaffected by Cold or Warm reset.</p> | x     |
| [38:32] | CECR | <p>Corrected error count, repeat. Incremented for the first countable error, which also records other syndrome for the error, and subsequently for each countable error that matches the recorded other syndrome.</p> <p>This field resets to an IMPLEMENTATION DEFINED which might be <b>UNKNOWN</b> on a Cold reset. If the reset value is <b>UNKNOWN</b>, then the value of this field remains <b>UNKNOWN</b> until software initializes it.</p> <p>Unaffected by Cold or Warm reset.</p>            | 7 {x} |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [31:28] | WAY      | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L1 Data Cache]</p> <ul style="list-style-type: none"> <li>Indicates which Tag RAM way or data RAM way detected the error. Upper 2 bits are unused.</li> </ul> <p>[L2 TLB]</p> <ul style="list-style-type: none"> <li>Indicates which RAM detected an error. The possible values are 0 (RAM 1) to 9 (RAM 10).</li> </ul> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which way detected the error. Upper 2 bits are unused.</li> </ul> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which way detected the error.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> | xxxx  |
| [27]    | SUBBANK  | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which subbank detected the error, valid for Instruction Data Cache. For Tag errors this field is zero.</li> </ul> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                                                                                                                                                         | x     |
| [26:25] | BANK     | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which L2 bank detected the error.</li> </ul> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which bank detected the error, valid for Instruction Data Cache. For Tag errors this field is zero.</li> </ul> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                                                                                                            | xx    |
| [24:21] | SUBARRAY | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which L2 data ECC granule detected the error. Upper bits are unused dependent on the ECC granule size.</li> </ul> <p>[L1 Data Cache]</p> <ul style="list-style-type: none"> <li>Indicates for L1 Data RAM which word had the error detected. For L1 Tag RAMs which bank had the error (0b0000: bank0 , 0b0001: bank1)</li> </ul> <p>Unaffected by Cold or Warm reset.</p>                                                                                                                                                                                      | xxxx  |

| Bits   | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset  |
|--------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [20:6] | INDEX | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <ul style="list-style-type: none"> <li>Indicates which index detected the error. Upper bits of the index are unused depending on the cache size.</li> </ul> <p>[L1 Data Cache]</p> <ul style="list-style-type: none"> <li>Indicates which index detected the error. Upper bits of the index are unused depending on the cache size</li> </ul> <p>[L2 TLB]</p> <ul style="list-style-type: none"> <li>Index of TLB RAM. Upper 4 bits are unused.</li> </ul> <p>[L1 Instruction Cache]</p> <ul style="list-style-type: none"> <li>Indicates which index detected the error. Upper bits of the index are unused depending on the cache size.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> | 15 {x} |

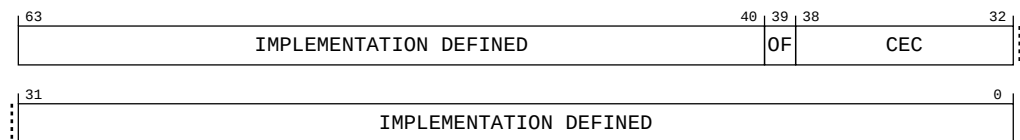
| Bits  | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|-------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [5:4] | ARRAY | <p>The encoding is dependent on the unit from which the error being recorded was detected. The possible values are:</p> <p>[L2 Cache]</p> <p>Indicates which array detected the error. The possible values are:</p> <ul style="list-style-type: none"> <li>0b00 L2 Tag RAM.</li> <li>0b01 L2 Data RAM.</li> <li>0b10 L2 TQ Data RAM.</li> <li>0b11 CHI Error.</li> </ul> <p>[L1 Data Cache]</p> <p>Indicates which array detected the error. The possible values are:</p> <ul style="list-style-type: none"> <li>00 LS Tag RAM 0.</li> <li>01 LS Tag RAM 1.</li> <li>10 LS Data RAM.</li> <li>11 LS Tag RAM 2.</li> </ul> <p>[L2 TLB]</p> <p>Indicates which array detected the error. The possible values are:</p> <ul style="list-style-type: none"> <li>00 Translation cache.</li> <li>01 GPT cache (when LEGACY_TZ_EN is 0).</li> <li>10 Reserved.</li> <li>11 Reserved.</li> </ul> <p>[L1 Instruction Cache]</p> <p>Indicates which array that detected the error, Data Array has higher priority. The possible values are:</p> <ul style="list-style-type: none"> <li>0b00 Tag.</li> <li>0b01 Data.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> | xx    |
| [3:0] | UNIT  | <p>Indicates the unit which detected the error. The possible values are:</p> <p><b>0b0001</b><br/>L1 Instruction Cache.</p> <p><b>0b0010</b><br/>L2 TLB.</p> <p><b>0b0100</b><br/>L1 Data Cache.</p> <p><b>0b1000</b><br/>L2 Cache.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | xxxx  |

When `ERRFR[FirstRecordOfNode(0)].CEC == '100' && ERRFR[FirstRecordOfNode(0)].RP == '0'`

**Figure B-140: ext\_err0misc0 bit assignments****Table B-293: ERR0MISC0 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                             | Reset  |
|---------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:48] | None | <b>IMPLEMENTATION DEFINED</b> syndrome.                                                                                                                                                 | 16 {x} |
| [47]    | OF   | Sticky overflow bit. Set to 1 when ERR<n>MISC0.CEC is incremented and wraps through zero.<br><br><b>0b0</b><br>Counter has not overflowed.<br><br><b>0b1</b><br>Counter has overflowed. | x      |
| [46:32] | CEC  | Corrected error count. Incremented for each Corrected error. It is <b>IMPLEMENTATION DEFINED</b> and might be UNPREDICTABLE whether Deferred and Uncorrected errors are counted.        | 15 {x} |
| [31:0]  | None | <b>IMPLEMENTATION DEFINED</b> syndrome.                                                                                                                                                 | 32 {x} |

When `ERRFR[FirstRecordOfNode(0)].CEC == '010' && ERRFR[FirstRecordOfNode(0)].RP == '0'`

**Figure B-141: ext\_err0misc0 bit assignments****Table B-294: ERR0MISC0 bit descriptions**

| Bits    | Name | Description                                                                                                                                                                             | Reset  |
|---------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:40] | None | <b>IMPLEMENTATION DEFINED</b> syndrome.                                                                                                                                                 | 24 {x} |
| [39]    | OF   | Sticky overflow bit. Set to 1 when ERR<n>MISC0.CEC is incremented and wraps through zero.<br><br><b>0b0</b><br>Counter has not overflowed.<br><br><b>0b1</b><br>Counter has overflowed. | x      |
| [38:32] | CEC  | Corrected error count. Incremented for each Corrected error. It is <b>IMPLEMENTATION DEFINED</b> and might be UNPREDICTABLE whether Deferred and Uncorrected errors are counted.        | 7 {x}  |
| [31:0]  | None | <b>IMPLEMENTATION DEFINED</b> syndrome.                                                                                                                                                 | 32 {x} |

When `ERRFR[FirstRecordOfNode(0)].CEC == '100' && ERRFR[FirstRecordOfNode(0)].RP == '1'`

Figure B-142: ext\_err0misc0 bit assignments

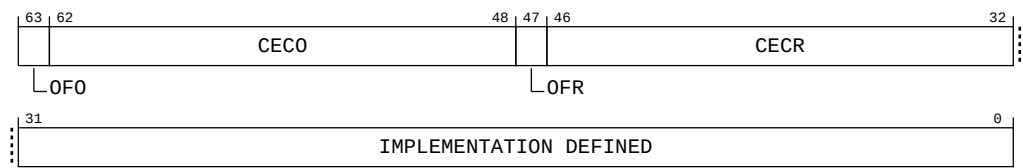


Table B-295: ERR0MISC0 bit descriptions

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                        | Reset  |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63]    | OFO  | Sticky overflow bit, other. Set to 1 when ERR<n>MISC0.CECO is incremented and wraps through zero.<br><br><b>0b0</b><br>Other counter has not overflowed.<br><br><b>0b1</b><br>Other counter has overflowed.                                                                                                                                                                        | x      |
| [62:48] | CECO | Corrected error count, other. Incremented for each countable error that is not accounted for by incrementing ERR<n>MISC0.CECR.                                                                                                                                                                                                                                                     | 15 {x} |
| [47]    | OFR  | Sticky overflow bit, repeat. Set to 1 when ERR<n>MISC0.CECR is incremented and wraps through zero.<br><br><b>0b0</b><br>Repeat counter has not overflowed.<br><br><b>0b1</b><br>Repeat counter has overflowed.                                                                                                                                                                     | x      |
| [46:32] | CECR | Corrected error count, repeat. Incremented for the first countable error, which also records other syndrome for the error, and subsequently for each countable error that matches the recorded other syndrome. Corrected errors are countable errors. It is <b>IMPLEMENTATION DEFINED</b> and might be UNPREDICTABLE whether Deferred and Uncorrected errors are countable errors. | 15 {x} |
| [31:0]  | None | <b>IMPLEMENTATION DEFINED</b> syndrome.                                                                                                                                                                                                                                                                                                                                            | 32 {x} |

When ERRFR[FirstRecordOfNode(0)].CEC == '010' && ERRFR[FirstRecordOfNode(0)].RP == '1'

Figure B-143: ext\_err0misc0 bit assignments

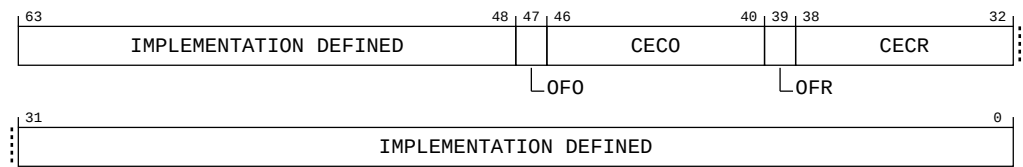


Table B-296: ERR0MISC0 bit descriptions

| Bits    | Name | Description                             | Reset  |
|---------|------|-----------------------------------------|--------|
| [63:48] | None | <b>IMPLEMENTATION DEFINED</b> syndrome. | 16 {x} |

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                        | Reset  |
|---------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [47]    | OFO  | Sticky overflow bit, other. Set to 1 when ERR<n>MISCO.CECO is incremented and wraps through zero.<br><br><b>0b0</b><br>Other counter has not overflowed.<br><br><b>0b1</b><br>Other counter has overflowed.                                                                                                                                                                        | x      |
| [46:40] | CECO | Corrected error count, other. Incremented for each countable error that is not accounted for by incrementing ERR<n>MISCO.CECC.                                                                                                                                                                                                                                                     | 7 {x}  |
| [39]    | OFR  | Sticky overflow bit, repeat. Set to 1 when ERR<n>MISCO.CECC is incremented and wraps through zero.<br><br><b>0b0</b><br>Repeat counter has not overflowed.<br><br><b>0b1</b><br>Repeat counter has overflowed.                                                                                                                                                                     | x      |
| [38:32] | CECC | Corrected error count, repeat. Incremented for the first countable error, which also records other syndrome for the error, and subsequently for each countable error that matches the recorded other syndrome. Corrected errors are countable errors. It is <b>IMPLEMENTATION DEFINED</b> and might be UNPREDICTABLE whether Deferred and Uncorrected errors are countable errors. | 7 {x}  |
| [31:0]  | None | <b>IMPLEMENTATION DEFINED</b> syndrome.                                                                                                                                                                                                                                                                                                                                            | 32 {x} |

## Access

Reads from ERR<n>MISCO return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



Note

These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

## Accessibility

Reads from ERR<n>MISCO return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



Note

These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| RAS       | 0x20   | ERR0MISC0 | None  |

This interface is accessible as follows:

RW

### B.8.6 ERR0MISC1, Error Record <n> Miscellaneous Register 1

**IMPLEMENTATION DEFINED** error syndrome register. The miscellaneous syndrome registers might contain:

- Information to locate where the error was detected.
- If the error was detected within a FRU, the identity of the FRU.
- A Corrected error counter or counters.
- Other state information not present in the corresponding status and address registers.

#### Configurations

ERRFR[FirstRecordOfNode(n)] describes the features implemented by the node that owns error record <n>. FirstRecordOfNode(n) is the index of the first error record owned by the same node as error record <n>. If the node owns a single record then FirstRecordOfNode(n) = n.

For **IMPLEMENTATION DEFINED** fields in ERR<n>MISC1, writing zero returns the error record to an initial quiescent state.

In particular, if any **IMPLEMENTATION DEFINED** syndrome fields might generate a Fault Handling or Error Recovery Interrupt request, writing zero is sufficient to deactivate the Interrupt request.

Fields that are read-only, nonzero, and ignore writes are compliant with this requirement.





Arm recommends that any IMPLEMENTATION DEFINED syndrome field that can generate a Fault Handling, Error Recovery, Critical, or IMPLEMENTATION DEFINED, interrupt request is disabled at Cold reset and is enabled by software writing an IMPLEMENTATION DEFINED nonzero value to an IMPLEMENTATION DEFINED field in ERRCTLR[FirstRecordOfNode(n)].

Attributes

Width

64

Component

RAS

Register offset

0x28

Access type

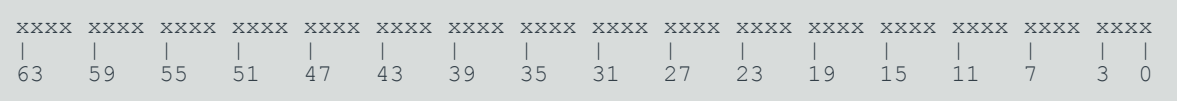
Read

R

Write

W

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-144: ext\_err0misc1 bit assignments

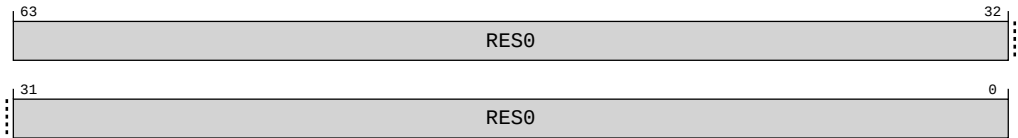


Table B-298: ERR0MISC1 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

Reads from ERR<n>MISC1 return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

Accessibility

Reads from ERR<n>MISC1 return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| RAS       | 0x28   | ERR0MISC1 | None  |

This interface is accessible as follows:

RW

### B.8.7 ERR0MISC2, Error Record <n> Miscellaneous Register 2

**IMPLEMENTATION DEFINED** error syndrome register. The miscellaneous syndrome registers might contain:

- Information to locate where the error was detected.
- If the error was detected within a FRU, the identity of the FRU.
- A Corrected error counter or counters.
- Other state information not present in the corresponding status and address registers.

#### Configurations

ERRFR[FirstRecordOfNode(n)] describes the features implemented by the node that owns error record <n>. FirstRecordOfNode(n) is the index of the first error record owned by the same node as error record <n>. If the node owns a single record then FirstRecordOfNode(n) = n.

For IMPLEMENTATION DEFINED fields in ERR<n>MISC2, writing zero returns the error record to an initial quiescent state.

In particular, if any IMPLEMENTATION DEFINED syndrome fields might generate a Fault Handling or Error Recovery Interrupt request, writing zero is sufficient to deactivate the Interrupt request.

Fields that are read-only, nonzero, and ignore writes are compliant with this requirement.

Arm recommends that if RAS System Architecture v1.1 is not implemented then ERR<n>MISC2 does not require zeroing to return the record to a quiescent state.



Note

Arm recommends that any IMPLEMENTATION DEFINED syndrome field that can generate a Fault Handling, Error Recovery, Critical, or IMPLEMENTATION DEFINED, interrupt request is disabled at Cold reset and is enabled by software writing an IMPLEMENTATION DEFINED nonzero value to an IMPLEMENTATION DEFINED field in ERRCTLR[FirstRecordOfNode(n)].

#### Attributes

**Width**

64

**Component**

RAS

**Register offset**

0x30

**Access type**

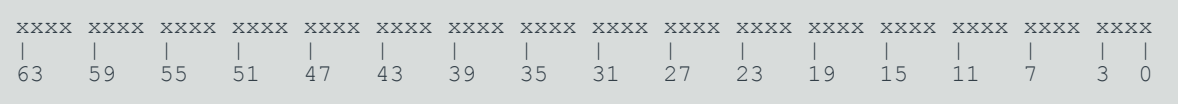
Read

R

Write

W

Reset value



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-145: ext\_err0misc2 bit assignments

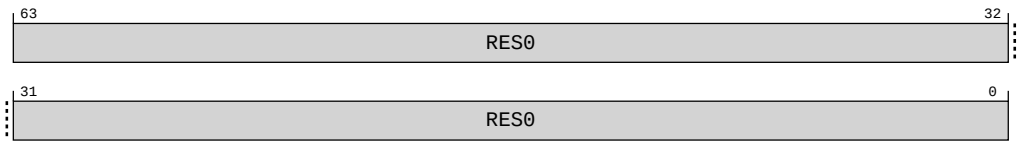


Table B-300: ERR0MISC2 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

Reads from ERR<n>MISC2 return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

Accessibility

Reads from ERR<n>MISC2 return an IMPLEMENTATION DEFINED value and writes have IMPLEMENTATION DEFINED behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| RAS       | 0x30   | ERR0MISC2 | None  |

This interface is accessible as follows:

RW

B.8.8 ERR0MISC3, Error Record <n> Miscellaneous Register 3

**IMPLEMENTATION DEFINED** error syndrome register. The miscellaneous syndrome registers might contain:

- Information to locate where the error was detected.
- If the error was detected within a FRU, the identity of the FRU.
- A Corrected error counter or counters.
- Other state information not present in the corresponding status and address registers.

If the node that owns error record n supports the RAS Timestamp Extension (ERRFR[FirstRecordOfNode(n)].TS != 0b00), then ERR<n>MISC3 contains the timestamp value for error record n when the error was detected. Otherwise the contents of ERR<n>MISC3 are IMPLEMENTATION DEFINED.

Configurations

ERRFR[FirstRecordOfNode(n)] describes the features implemented by the node that owns error record <n>. FirstRecordOfNode(n) is the index of the first error record owned by the same node as error record <n>. If the node owns a single record then FirstRecordOfNode(n) = n.

For IMPLEMENTATION DEFINED fields in ERR<n>MISC3, writing zero returns the error record to an initial quiescent state.

In particular, if any IMPLEMENTATION DEFINED syndrome fields might generate a Fault Handling or Error Recovery Interrupt request, writing zero is sufficient to deactivate the Interrupt request.

Fields that are read-only, nonzero, and ignore writes are compliant with this requirement.

Arm recommends that if RAS System Architecture v1.1 is not implemented then ERR<n>MISC3 does not require zeroing to return the record to a quiescent state.



Arm recommends that any IMPLEMENTATION DEFINED syndrome field that can generate a Fault Handling, Error Recovery, Critical, or IMPLEMENTATION DEFINED, interrupt request is disabled at Cold reset and is enabled by software writing an IMPLEMENTATION DEFINED nonzero value to an IMPLEMENTATION DEFINED field in ERRCTLR[FirstRecordOfNode(n)].

Attributes

Width

64

Component

RAS

Register offset

0x38

Access type

Read

R

Write

W

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

Bit descriptions

Figure B-146: ext\_err0misc3 bit assignments

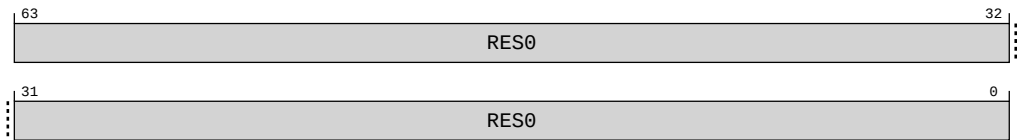


Table B-302: ERR0MISC3 bit descriptions

| Bits   | Name | Description | Reset |
|--------|------|-------------|-------|
| [63:0] | RES0 | Reserved    | RES0  |

Access

Reads from ERR<n>MISC3 return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

Accessibility

Reads from ERR<n>MISC3 return an **IMPLEMENTATION DEFINED** value and writes have **IMPLEMENTATION DEFINED** behavior.

If the Common Fault Injection Mechanism is implemented by the node that owns this error record, and ERRPFGF[FirstRecordOfNode(n)].MV is 1, then some parts of this register are read/write when ext-ERR<n>STATUS.MV is 0. See ext-ERR<n>PFGF.MV for more information.

For other parts of this register, or if the Common Fault Injection Mechanism is not implemented, then Arm recommends that:

- Miscellaneous syndrome for multiple errors, such as a corrected error counter, is read/write.
- When ext-ERR<n>STATUS.MV is 1, the miscellaneous syndrome specific to the most recently recorded error ignores writes.



These recommendations allow a counter to be reset in the presence of a persistent error, while preventing specific information, such as that identifying a FRU, from being lost if an error is detected while the previous error is being logged.

| Component | Offset | Instance  | Range |
|-----------|--------|-----------|-------|
| RAS       | 0x38   | ERR0MISC3 | None  |

This interface is accessible as follows:

RW

### B.8.9 ERR0PFGF, Error Record <n> Pseudo-fault Generation Feature Register

Defines which common architecturally-defined fault generation features are implemented.

#### Configurations

ext-ERR<n>FR describes the features implemented by the node.

#### Attributes

##### Width

64

##### Component

RAS

##### Register offset

0x800

##### Access type

RO

##### Reset value

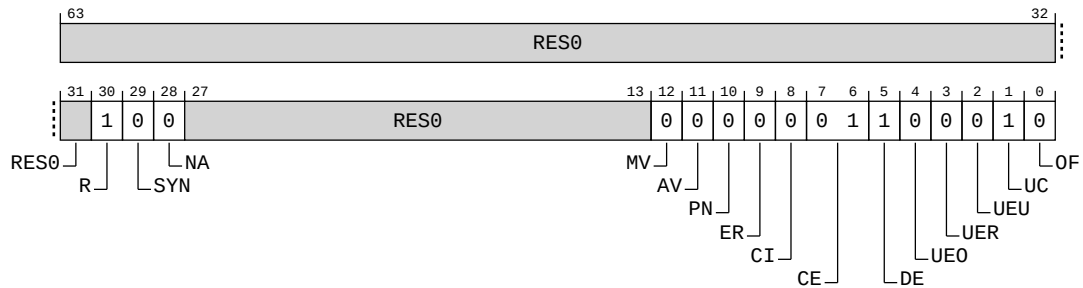
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | x100 | xxxx | xxxx | xxxx | xxx0 | 0000 | 0110 | 0010 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



**Note**

Where the reset reads xxxx, see individual bits.

## Bit descriptions

**Figure B-147: ext\_errOpfgf bit assignments****Table B-304: ERR0PFGF bit descriptions**

| Bits    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|---------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [63:31] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0  |
| [30]    | R    | Restartable. Support for Error Generation Counter restart mode.<br><b>0b1</b><br>Error Generation Counter restart mode is implemented and is controlled by ext-ERR<n>PFGCTL.R.                                                                                                                                                                                                                                                                                                                                                | 0b1   |
| [29]    | SYN  | Syndrome. Fault syndrome injection.<br><b>0b0</b><br>When an injected error is recorded, the node sets ext-ERR<n>STATUS.{IERR, SERR} to IMPLEMENTATION DEFINED values. ext-ERR<n>STATUS.{IERR, SERR} are <b>UNKNOWN</b> when ext-ERR<n>STATUS.V is 0.                                                                                                                                                                                                                                                                         | 0b0   |
| [28]    | NA   | No access required. Defines whether this component fakes detection of the error on an access to the component or spontaneously in the fault injection state.<br><b>0b0</b><br>The component fakes detection of the error on an access to the component.                                                                                                                                                                                                                                                                       | 0b0   |
| [27:13] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RES0  |
| [12]    | MV   | Miscellaneous syndrome.<br>Defines whether software can control all or part of the syndrome recorded in the ERR<n>MISC<m> registers when an injected error is recorded.<br><b>0b0</b><br>When an injected error is recorded, the node might update the ERR<n>MISC<m> registers: <ul style="list-style-type: none"> <li>If any syndrome is recorded by the node in the ERR&lt;n&gt;MISC&lt;m&gt; registers, then ext-ERR&lt;n&gt;STATUS.MV is set to 1.</li> <li>Otherwise, ext-ERR&lt;n&gt;STATUS.MV is unchanged.</li> </ul> | 0b0   |

| Bits  | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [11]  | AV   | <p>Address syndrome. Defines whether software can control the address recorded in ext-ERR&lt;n&gt;ADDR when an injected error is recorded.</p> <p><b>0b0</b></p> <p>When an injected error is recorded, the node might record an address in ext-ERR&lt;n&gt;ADDR. If an address is recorded in ext-ERR&lt;n&gt;ADDR, then ext-ERR&lt;n&gt;STATUS.AV is set to 1. Otherwise, ext-ERR&lt;n&gt;ADDR and ext-ERR&lt;n&gt;STATUS.AV are unchanged.</p> | 0b0   |
| [10]  | PN   | <p>Poison flag. Describes how the fault generation feature of the node sets the ext-ERR&lt;n&gt;STATUS.PN status flag.</p> <p><b>0b0</b></p> <p>When an injected error is recorded, the node sets ext-ERR&lt;n&gt;STATUS.PN to 0.</p>                                                                                                                                                                                                             | 0b0   |
| [9]   | ER   | <p>Error Reported flag. Describes how the fault generation feature of the node sets the ext-ERR&lt;n&gt;STATUS.ER status flag.</p> <p><b>0b0</b></p> <p>When an injected error is recorded, the node sets ext-ERR&lt;n&gt;STATUS.ER according to the architecture-defined rules for setting the ER field.</p>                                                                                                                                     | 0b0   |
| [8]   | CI   | <p>Critical Error flag. Describes how the fault generation feature of the node sets the ext-ERR&lt;n&gt;STATUS.CI status flag.</p> <p><b>0b0</b></p> <p>The node does not support this type of flag. This behavior replaces the architecture-defined rules for setting the CI bit.</p>                                                                                                                                                            | 0b0   |
| [7:6] | CE   | <p>Corrected Error generation. Describes the types of Corrected error that the fault generation feature of the node can generate.</p> <p><b>0b01</b></p> <p>The fault generation feature of the node allows generation of a non-specific Corrected error, that is, a Corrected error that is recorded by setting ext-ERR&lt;n&gt;STATUS.CE to 0b10.</p>                                                                                           | 0b01  |
| [5]   | DE   | <p>Deferred Error generation. Describes whether the fault generation feature of the node can generate Deferred errors.</p> <p><b>0b1</b></p> <p>The fault generation feature of the node allows generation of Deferred errors.</p>                                                                                                                                                                                                                | 0b1   |
| [4]   | UEO  | <p>Latent or Restartable Error generation. Describes whether the fault generation feature of the node can generate Latent or Restartable errors.</p> <p><b>0b0</b></p> <p>The fault generation feature of the node does not generate Latent or Restartable errors.</p>                                                                                                                                                                            | 0b0   |
| [3]   | UER  | <p>Signaled or Recoverable Error generation. Describes whether the fault generation feature of the node can generate Signaled or Recoverable errors.</p> <p><b>0b0</b></p> <p>The fault generation feature of the node does not generate Signaled or Recoverable errors.</p>                                                                                                                                                                      | 0b0   |
| [2]   | UEU  | <p>Unrecoverable Error generation. Describes whether the fault generation feature of the node can generate Unrecoverable errors.</p> <p><b>0b0</b></p> <p>The fault generation feature of the node does not generate Unrecoverable errors.</p>                                                                                                                                                                                                    | 0b0   |
| [1]   | UC   | <p>Uncontainable Error generation. Describes whether the fault generation feature of the node can generate Uncontainable errors.</p> <p><b>0b1</b></p> <p>The fault generation feature of the node allows generation of Uncontainable errors.</p>                                                                                                                                                                                                 | 0b1   |

| Bits | Name | Description                                                                                                                                                                                                                                                                         | Reset |
|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [0]  | OF   | Overflow flag. Describes how the fault generation feature of the node sets the ext-ERR<n>STATUS.OF status flag.<br><br><b>0b0</b><br><br>When an injected error is recorded, the node sets ext-ERR<n>STATUS.OF according to the architecture-defined rules for setting the OF field | 0b0   |

Accessibility

| Component | Offset | Instance | Range |
|-----------|--------|----------|-------|
| RAS       | 0x800  | ERR0PFGF | None  |

This interface is accessible as follows:

RO

B.8.10 ERR0PFGCTL, Error Record <n> Pseudo-fault Generation Control Register

Enables controlled fault generation.

Configurations

ext-ERR<n>PFGF describes the Common Fault Injection features implemented by the node.

ext-ERR<n>FR describes the features implemented by the node.

Attributes

Width

64

Component

RAS

Register offset

0x808

Access type

RW

Reset value

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | xxxx | 00xx | xxxx | xxxx | xxxx | xxx1 | x0x0 | 000x | xx00 |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 63   | 59   | 55   | 51   | 47   | 43   | 39   | 35   | 31   | 27   | 23   | 19   | 15   | 11   | 7    | 3 0  |



Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure B-148: ext\_errOpfgctl bit assignments

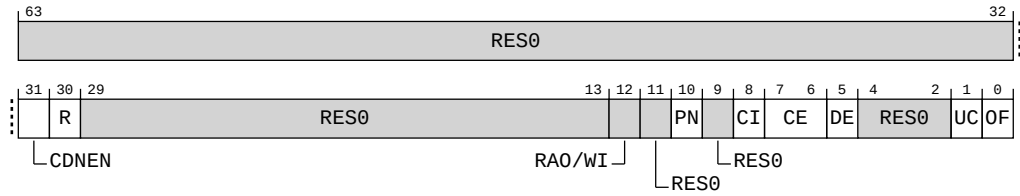


Table B-306: ERR0PFGCTL bit descriptions

| Bits    | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Reset  |
|---------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [63:32] | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0   |
| [31]    | CDNEN  | Countdown Enable. Controls transfers from the value that is held in the ERXPFGCDN_EL1 into the Error Generation Counter and enables this counter.<br><br><b>0b0</b><br>The Error Generation Counter is disabled.<br><br><b>0b1</b><br>The Error Generation Counter is enabled. On a write of 0b1 to this bit, the Error Generation Counter is set to ERXPFGCDN_EL1.CDN.<br><br>Cold reset only. Unaffected by Warm reset | 0b0    |
| [30]    | R      | Restart. Controls whether, upon reaching zero, the Error Generation Counter restarts from the ERXPFGCDN_EL1 value or stops.<br><br><b>0b0</b><br>On reaching 0, the Error Generation Counter will stop.<br><br><b>0b1</b><br>On reaching 0, the Error Generation Counter is set to ERXPFGCDN_EL1.CDN.<br><br>Cold reset only. Unaffected by Warm reset                                                                   | 0b0    |
| [29:13] | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0   |
| [12]    | RAO/WI | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RAO/WI |
| [11]    | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0   |
| [10]    | PN     | Poison flag. The value that is written to AArch64-ERXSTATUS.PN when an injected error is recorded.<br><br><b>0b00</b><br>AArch64-ERXSTATUS.PN is set to 0 when an injected error is recorded.<br><br><b>0b01</b><br>AArch64-ERXSTATUS.PN is set to 1 when an injected error is recorded.                                                                                                                                 | 0b0    |
| [9]     | RES0   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | RES0   |
| [8]     | CI     | Critical Error flag. The value that is written to AArch64-ERXSTATUS.CI when an injected error is recorded.<br><br><b>0b0</b><br>AArch64-ERXSTATUS.CI is set to 0 when an injected error is recorded.<br><br><b>0b1</b><br>AArch64-ERXSTATUS.CI is set to 1 when an injected error is recorded.                                                                                                                           | 0b0    |

| Bits  | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| [7:6] | CE   | Corrected Error generation enable. Controls the type of Corrected Error condition that might be generated. The possible values are:<br><br><b>0b00</b><br>No error of this type will be generated.<br><br><b>0b01</b><br>A non-specific Corrected Error, that is, a Corrected Error that is recorded as ERXSTATUS_EL1.CE == 0b10, might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset | 0b00  |
| [5]   | DE   | Deferred Error generation enable. The possible values are:<br><br><b>0b0</b><br>No error of this type will be generated.<br><br><b>0b1</b><br>An error of this type might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                               | 0b0   |
| [4:2] | RES0 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RES0  |
| [1]   | UC   | Uncontainable Error generation enable. The possible values are:<br><br><b>0b0</b><br>No error of this type will be generated.<br><br><b>0b1</b><br>An error of this type might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                          | 0b0   |
| [0]   | OF   | Uncontainable Error generation enable. The possible values are:<br><br><b>0b0</b><br>No error of this type will be generated.<br><br><b>0b1</b><br>An error of this type might be generated when the Error Generation Counter decrements to zero.<br><br>Cold reset only. Unaffected by Warm reset                                                                                                                                                          | 0b0   |

### Accessibility

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| RAS       | 0x808  | ERROPFGCTL | None  |

This interface is accessible as follows:

RW

### B.8.11 ERR0PFGCDN, Error Record <n> Pseudo-fault Generation Countdown Register

Generates one of the errors enabled in the corresponding ext-ERR<n>PFGCTL register.

#### Configurations

ext-ERR<n>FR describes the features implemented by the node.

#### Attributes

##### Width

64

##### Component

RAS

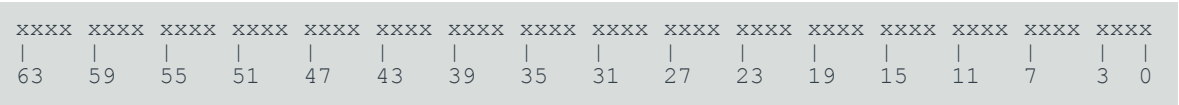
##### Register offset

0x810

##### Access type

RW

##### Reset value



Note

Where the reset reads xxxx, see individual bits.

#### Bit descriptions

Figure B-149: ext\_err0pfgcdn bit assignments

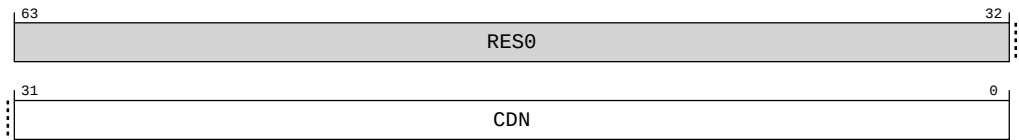


Table B-308: ERR0PFGCDN bit descriptions

| Bits    | Name | Description | Reset |
|---------|------|-------------|-------|
| [63:32] | RES0 | Reserved    | RES0  |

| Bits   | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Reset  |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| [31:0] | CDN  | <p>Countdown value.</p> <p>This field is copied to Error Generation Counter when either:</p> <ul style="list-style-type: none"> <li>Software writes ERXPFGCTL_EL1.CDNEN with 1.</li> <li>The Error Generation Counter decrements to zero and ERXPFGCTL_EL1.R == 0b1.</li> </ul> <p>Unaffected by Cold or Warm reset.</p> <p><b>Note:</b><br/>The current Error Generation Counter value is not visible to software.</p> | 32 {x} |

### Accessibility

| Component | Offset | Instance   | Range |
|-----------|--------|------------|-------|
| RAS       | 0x810  | ERROPFGCDN | None  |

This interface is accessible as follows:

RW

# Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant



export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

# Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

## Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

### Product completeness status

The information in this document is Final, that is for a developed product.

### Product revision status

This product is rOp2, which indicates the revision status of the product described in this manual, where:

- r (value)**Identifies the major revision of the product, for example, r1.
- p (value)**Identifies the minor revision or modification status of the product, for example, p2.

## Revision history

These sections can help you understand how the document has changed over time.

### Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

#### Document history

| Issue   | Date            | Confidentiality  | Change                                |
|---------|-----------------|------------------|---------------------------------------|
| 0002-06 | 22 May 2025     | Non-Confidential | First release for rOp2                |
| 0001-05 | 29 May 2024     | Non-Confidential | Second early access release for rOp1  |
| 0001-04 | 31 August 2023  | Confidential     | First early access release for rOp1   |
| 0000-03 | 5 April 2023    | Confidential     | First limited access release for rOp0 |
| 0000-02 | 31 January 2023 | Confidential     | First Beta release for rOp0           |

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in [Document release information](#) on page 882.

**Table 2: Differences between issue 0000-01 and issue 0000-02**

| Change                                   | Location |
|------------------------------------------|----------|
| First Confidential Beta release for r0p0 | -        |

**Table 3: Differences between issue 0000-02 and issue 0000-03**

| Change                                                              | Location                                                                                                                                                                 |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| First Confidential limited access release for r0p0                  | -                                                                                                                                                                        |
| Updates to Hardware data prefetcher                                 | <a href="#">7.5 Data prefetching</a> on page 66                                                                                                                          |
| Updates and corrections in L2 cache encodings                       | <a href="#">9.2 L2 cache encodings</a> on page 79                                                                                                                        |
| Architecturally defined registers listed in register summary tables | <ul style="list-style-type: none"> <li>• <a href="#">A. AArch64 System registers</a> on page 165</li> <li>• <a href="#">B. External registers</a> on page 620</li> </ul> |

**Table 4: Differences between issue 0000-03 and issue 0001-04**

| Change                                                       | Location                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| First Confidential early access release for r0p1             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Minor editorial updates in registers to reflect r0p1 changes | <ul style="list-style-type: none"> <li>• <a href="#">A.1.2 IMP_CPUPPMCR_EL3, Global Performance and Power Management Configuration Register</a> on page 167</li> <li>• <a href="#">A.15.3 PMSIDR_EL1, Sampling Profiling ID Register</a> on page 613</li> <li>• <a href="#">B.1.12 COREROM_PIDR3, Core ROM table Peripheral Identification Register 3</a> on page 635</li> <li>• <a href="#">B.2.1 CPUPPMCR, Global Performance and Power Management Configuration Register</a> on page 642</li> <li>• <a href="#">B.3.21 PMPIDR3, Performance Monitors Peripheral Identification Register 3</a> on page 695</li> <li>• <a href="#">B.4.3 MIDR_EL1, Main ID Register</a> on page 708</li> <li>• <a href="#">B.4.20 EDCIDR3, External Debug Component Identification Register 3</a> on page 733</li> <li>• <a href="#">B.6.17 AMIIDR, Activity Monitors Implementation Identification Register</a> on page 767</li> <li>• <a href="#">B.6.24 AMPIDR3, Activity Monitors Peripheral Identification Register 3</a> on page 777</li> <li>• <a href="#">B.7.37 TRCCIDR3, Component Identification Register 3</a> on page 837</li> </ul> |

**Table 5: Differences between Issue 0001-04 and 0001-05**

| Change                                                | Location                                                                            |
|-------------------------------------------------------|-------------------------------------------------------------------------------------|
| Second Non-Confidential early access release for r0p1 | -                                                                                   |
| Updated product name                                  | Throughout book                                                                     |
| Updated register bit description                      | <a href="#">B.3.1 PMPCSSR, Snapshot Program Counter Sample Register</a> on page 654 |
| Updated event name                                    | <a href="#">17.1 Performance monitors events</a> on page 119                        |
| Removed unnecessary data                              | <a href="#">4.5.2 Performance Defined Power</a> on page 47                          |
| Updated table                                         | <a href="#">9.2 L2 cache encodings</a> on page 79                                   |
| Added text on WFI instructions                        | <a href="#">4.4.6 Warm reset mode</a> on page 45                                    |

**Table 6: Differences between Issue 0001-05 and 0002-06**

| Change                                                                           | Location                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| First Non-Confidential access release for r0p2                                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Editorial changes                                                                | Throughout document                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Caution and Note added; clarified use of WARM_RST mode and added WFI information | <a href="#">4.4.6 Warm reset mode</a> on page 45                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Updated description of conflict aborts                                           | <a href="#">5.6 Responses</a> on page 54                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Added new table for L2 Victim RAM                                                | <a href="#">9.2.4 L2 Victim RAM returned data</a> on page 86                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Clarified the cache protection capabilities of core RAMs                         | <a href="#">10.1 Cache protection behavior</a> on page 88                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Updated CoreSight peripheral ID                                                  | <a href="#">16.6 CoreSight component identification</a> on page 113                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Updated CTI register identification values                                       | <a href="#">16.7 CTI register identification values</a> on page 113                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Updated INST_PEC, L1D_CACHE_HWPRF, and L3D_CACHE_RW PMU event description        | <a href="#">17.1 Performance monitors events</a> on page 119                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Removed PC_WRITE_RETIRED, CSDB_SPEC, and BR_IMMED_TAKEN_RETIRED PMU events       | <a href="#">17.1 Performance monitors events</a> on page 119                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Editorial revisions                                                              | <a href="#">A. AArch64 System registers</a> on page 165                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Minor editorial updates in registers to reflect r0p2 changes.                    | <ul style="list-style-type: none"> <li><a href="#">B.4.3 MIDR_EL1, Main ID Register</a> on page 708</li> <li><a href="#">B.6.17 AMIIDR, Activity Monitors Implementation Identification Register</a> on page 767</li> <li><a href="#">B.6.23 AMPIDR2, Activity Monitors Peripheral Identification Register 2</a> on page 775</li> <li><a href="#">B.1.11 COREROM_PIDR2, Core ROM table Peripheral Identification Register 2</a> on page 634</li> <li><a href="#">B.4.15 EDPIDR2, External Debug Peripheral Identification Register 2</a> on page 727</li> <li><a href="#">B.7.32 TRCPIDR2, Peripheral Identification Register 2</a> on page 830</li> <li><a href="#">B.3.20 PMPIDR2, Performance Monitors Peripheral Identification Register 2</a> on page 693</li> </ul> |
| Removed DBGBCR<n> register descriptions                                          | <a href="#">A.3 AArch64 Debug registers summary</a> on page 282                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Removed DBGWCR<n> register descriptions                                          | <a href="#">A.3 AArch64 Debug registers summary</a> on page 282                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Removed DBGBVR<n> register descriptions                                          | <a href="#">A.3 AArch64 Debug registers summary</a> on page 282                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Removed DBGWVR<n> register descriptions                                          | <a href="#">A.3 AArch64 Debug registers summary</a> on page 282                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Updated CLIDR_EL1 bit descriptions                                               | <a href="#">A.5.20 CLIDR_EL1, Cache Level ID Register</a> on page 342                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Included PMEVCNTR6_ELO - PMEVCNTR30 register descriptions                        | <a href="#">A.7 AArch64 Performance Monitors registers summary</a> on page 352                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Included PMEVTYPER6_ELO - PMEVTYPER30_ELO register descriptions                  | <a href="#">A.7 AArch64 Performance Monitors registers summary</a> on page 352                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Removed TRCRSCTLR<n> register descriptions                                       | <a href="#">A.12 AArch64 Trace unit registers summary</a> on page 508                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Editorial revisions                                                              | <a href="#">B. External registers</a> on page 620                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Removed DBGBCR<n> register descriptions                                          | <a href="#">B.4 External Debug registers summary</a> on page 702                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Removed DBGWCR<n> register descriptions                                          | <a href="#">B.4 External Debug registers summary</a> on page 702                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Removed DBGBVR<n> register descriptions                                          | <a href="#">B.4 External Debug registers summary</a> on page 702                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Removed DBGWVR<n> register descriptions                                          | <a href="#">B.4 External Debug registers summary</a> on page 702                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Updated PPM registers access type to RW                                          | <a href="#">B.2 External PPM registers summary</a> on page 641                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

# Conventions

The following subsections describe conventions used in Arm documents.

## Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

## Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

| Convention                 | Use                                                                                                                                                                                                            |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>italic</i>              | Citations.                                                                                                                                                                                                     |
| <b>bold</b>                | Terms in descriptive lists, where appropriate.                                                                                                                                                                 |
| monospace                  | Text that you can enter at the keyboard, such as commands, file and program names, and source code.                                                                                                            |
| monospace <u>underline</u> | A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.                                                                                |
| <and>                      | Encloses replaceable terms for assembler syntax where they appear in code or code fragments.<br><br>For example:<br><div>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</div>              |
| SMALL CAPITALS             | Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> . |



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.



This information is important and needs your attention.



This information might help you perform a task in an easier, better, or faster way.



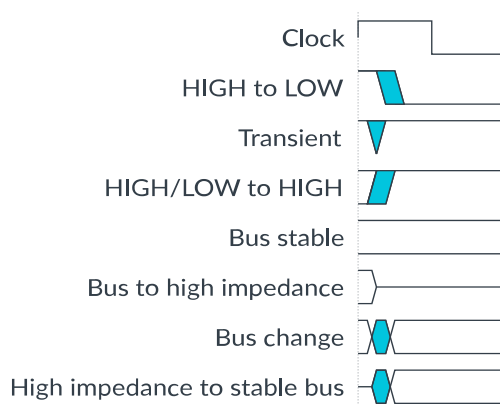
This information reminds you of something important relating to the current content.

## Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

**Figure 1: Key to timing diagram conventions**



## Signals

The signal conventions are:

### Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

**Lowercase n**

At the start or end of a signal name, n denotes an active-LOW signal.

# Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on [developer.arm.com/documentation](https://developer.arm.com/documentation).

Confidential documents are only available to licensees, when logged in. Each document link in the tables below provides direct access to the online version of the document.

| Arm product resources                                                                      | Document ID | Confidentiality  |
|--------------------------------------------------------------------------------------------|-------------|------------------|
| <a href="#">Arm® CoreSight™ Architecture Specification v3.0</a>                            | IHI 0029    | Non-Confidential |
| <a href="#">Arm® CoreSight™ ELA-600 Embedded Logic Analyzer Technical Reference Manual</a> | 101088      | Non-Confidential |
| <a href="#">Arm®Cortex®-X925 (MP210) Release Note</a>                                      | 110429      | Confidential     |
| <a href="#">Arm® Cortex®-X925 Core Configuration and Integration Manual</a>                | 102808      | Confidential     |
| <a href="#">Arm® Cortex®-X925 Core Cryptographic Extension Technical Reference Manual</a>  | 102809      | Non-Confidential |
| <a href="#">Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual</a>                   | 102547      | Non-Confidential |
| <a href="#">Arm® DynamIQ™ Shared Unit-120 Configuration and Integration Manual</a>         | 102548      | Confidential     |

| Arm architecture and specifications                                                                                    | Document ID | Confidentiality  |
|------------------------------------------------------------------------------------------------------------------------|-------------|------------------|
| <a href="#">AMBA® CHI Architecture Specification</a>                                                                   | IHI 0050    | Non-Confidential |
| <a href="#">Arm® Architecture Reference Manual for A-profile architecture</a>                                          | DDI 0487    | Non-Confidential |
| <a href="#">Arm® Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4</a> | IHI 0069    | Non-Confidential |
| <a href="#">Arm® Memory System Resource Partitioning and Monitoring (MPAM) System Component Specification</a>          | IHI 0099    | Non-Confidential |
| <a href="#">Arm® Reliability, Availability, and Serviceability (RAS) System Architecture</a>                           | IHI 0100    | Non-Confidential |
| <a href="#">Arm® CoreSight™ Architecture Specification v3.0</a>                                                        | IHI 0029    | Non-Confidential |