# Arm Cortex-A510 Core (MP164)

# Software Developer Errata Notice

**Date of issue:** December 19, 2023

**Non-Confidential**

**Document version:** 16.0

**Document ID:** SDEN-1873361

This document contains all known errata since the r0p0 release of the product.

# Non-confidential proprietary notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

## Confidentiality status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Cortex-A510 Core (MP164), 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**.

## 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.

If you find offensive language in this document, please email **terms@arm.com**.

# Contents

# r1p1 implementation fixes

Note the following errata might be fixed in some implementations of r1p1. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

| | | |
|---|---|---|
| REVIDR_EL1[24] | 2454944 | Unmodified cache line might be written back to memory |
| REVIDR_EL1[25] | 2658417 | BFMMLA or VMMLA instructions might produce incorrect result |

Note that there is no change to the MIDR_EL1 which remains at r1p1 but the REVIDR is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

# r1p0 implementation fixes

Note the following errata might be fixed in some implementations of r1p0. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

| | | |
|---|---|---|
| REVIDR_EL1[15] | 2217821 | Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock |
| REVIDR_EL1[18] | 2218950 | Aliased loads and stores might cause an ordering violation |
| REVIDR_EL1[19] | 2184257 | Normal execution can result in data corruption |
| REVIDR_EL1[22] | 2250311 | Asynchronous exceptions while MPMM is active might corrupt processor state |

Note that there is no change to the MIDR_EL1 which remains at r1p0 but the REVIDR is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

# r0p3 implementation fixes

Note the following errata might be fixed in some implementations of r0p3. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

| | | |
|---|---|---|
| REVIDR_EL1[15] | 2217821 | Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock |
| REVIDR_EL1[18] | 2218950 | Aliased loads and stores might cause an ordering violation |
| REVIDR_EL1[19] | 2184257 | Normal execution can result in data corruption |

Note that there is no change to the MIDR_EL1 which remains at r0p3 but the REVIDR is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

# r0p2 implementation fixes

Note the following errata might be fixed in some implementations of r0p2. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

| | | |
|---|---|---|
| REVIDR_EL1[15] | 2217821 | Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock |
| REVIDR_EL1[18] | 2218950 | Aliased loads and stores might cause an ordering violation |
| REVIDR_EL1[19] | 2184257 | Normal execution can result in data corruption |
| REVIDR_EL1[20] | 2184257 | Normal execution can result in data corruption. Note : either of the REVIDR_EL1 bit 19 or 20 indicates that this erratum is fixed |

Note that there is no change to the MIDR_EL1 which remains at r0p2 but the REVIDR is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

## r0p1 implementation fixes

Note the following errata might be fixed in some implementations of r0p1. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

| | | |
|---|---|---|
| REVIDR_EL1[12] | 1975996 | Non-L1-allocating stores might cause data corruption |

Note that there is no change to the MIDR_EL1 which remains at r0p1 but the REVIDR is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

## r0p0 implementation fixes

Note the following errata might be fixed in some implementations of r0p0. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

**When REVIDR_EL1[11] = 0:**

| | | |
|---|---|---|
| REVIDR_EL1[0] | 1914417 | Coherency may be lost |
| REVIDR_EL1[1] | 1921103 | TLBI completion might not be guaranteed by DSB |
| REVIDR_EL1[3] | 1927171 | Mismatched memory attributes might cause deadlock |
| REVIDR_EL1[5] | 1943861 | Acquire/release semantics violated on store atomic instructions |
| REVIDR_EL1[7] | 1955641 | Core in FULL_RET might cause data corruption |
| REVIDR_EL1[8] | 1970520 | Secure Physical Address Cache Invalidate all DVM operation does not invalidate Non-Secure lines in L1 I-Cache |
| | 1968719 | Snoops might lead to a deadlock |
| REVIDR_EL1[10] | 1996706 | A deadlock might occur during or after WFI or WFE |

**When REVIDR_EL1[11] = 1:**

| | | |
|---|---|---|
| REVIDR_EL1[0] | 1914417 | Coherency may be lost |
| | 1921103 | TLBI completion might not be guaranteed by DSB |
| | 1943861 | Acquire/release semantics violated on store atomic instructions |
| | 1968719 | Snoops might lead to a deadlock |
| | 1970520 | Secure Physical Address Cache Invalidate all DVM operation does not invalidate Non-Secure lines in L1 I-Cache |
| | 1996706 | A deadlock might occur during or after WFI or WFE |
| REVIDR_EL1[1] | 1955641 | Core in FULL_RET might cause data corruption |
| REVIDR_EL1[2] | 1927171 | Mismatched memory attributes might cause deadlock |
| REVIDR_EL1[4] | 1981843 | A store exclusive to shareable memory other than Normal Inner Write-Back, Outer Write-Back Cacheable might cause a hang |
| REVIDR_EL1[5] | 2007174 | Heavy store traffic might lead to a DSB not completing on the same core or another core |

Note that there is no change to the MIDR_EL1 which remains at r0p0 but the REVIDR is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

# Introduction

## Scope

This document describes errata categorized by level of severity. Each description includes:

- The current status of the erratum.
- Where the implementation deviates from the specification and the conditions required for erroneous behavior to occur.
- The implications of the erratum with respect to typical applications.
- The application and limitations of a workaround where possible.

## Categorization of errata

Errata are split into three levels of severity and further qualified as common or rare:

| | |
|---|---|
| **Category A** | A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications. |
| **Category A (Rare)** | A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage. |
| **Category B** | A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications. |
| **Category B (Rare)** | A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage. |
| **Category C** | A minor error. |

# Change Control

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The **errata summary table** identifies errata that have been fixed in each product revision.

December 19, 2023: Changes in document version v16.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| **2971420** | New | Programmer | Category B | Data corruption or deadlock might happen if TRBE is enabled |
| **3057514** | New | Programmer | Category B | Deferred error might become uncontainable when BROADCASTMTE is set |
| **3117295** | New | Programmer | Category B | A speculatively executed unprivileged load might leak data from a privileged via side channel |
| **2834345** | New | Programmer | Category C | Direct access to internal memory might not be reliable |
| **2856823** | New | Programmer | Category C | Speculative dirty bit hardware update might happen for store operation |
| **2867016** | New | Programmer | Category C | An uncontainable error might deadlock the cluster |
| **2867018** | New | Programmer | Category C | Error record registers indicate incorrect feature support in configurations without cache protection |
| **2878694** | New | Programmer | Category C | LDG or MTE checked load/store might fail to detect poisoned data |
| **2879976** | New | Programmer | Category C | Unmodified cache line might be written back to memory |
| **2971621** | New | Programmer | Category C | Unmodified pagetable cachelines might be written back to memory |
| **2976328** | New | Programmer | Category C | Store data might be lost when a correctable error is detected in the L1 data cache |

September 30, 2022: Changes in document version v15.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| **2666669** | New | Programmer | Category B | Data corruption might occur during core powerdown |
| **2675636** | New | Programmer | Category B | An asynchronous MTE check might not observe correct memory ordering |
| **2684597** | New | Programmer | Category B | A core might deadlock during powerdown if TRBE is enabled |
| **2724177** | New | Programmer | Category B | Deferred error might become uncontainable |
| **2679270** | New | Programmer | Category C | External aborts reporting can not be disabled |
| **2688792** | New | Programmer | Category C | PMU event L3D_CACHE_LMISS_RD might be inaccurate |
| **2690487** | New | Programmer | Category C | Some architectural PMU events are not always available to trace unit |
| **2710402** | New | Programmer | Category C | Read value of IMP_CPUCFR_EL1 might be incorrect |
| **2713359** | New | Programmer | Category C | ERR2STATUS.UET field might be incorrect |
| **2718749** | New | Programmer | Category C | Cache debug target for L2 Data RAM may not record correct data |
| **2736240** | New | Programmer | Category C | ERR2STATUS might be incorrect |

June 27, 2022: Changes in document version v14.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2008766 | Updated | Programmer | Category B | RAS errors during core power down might cause a deadlock |
| 2658417 | New | Programmer | Category B | BFMMLA or VMMLA instructions might produce incorrect result |
| 2618004 | New | Programmer | Category C | LDST_SPEC event might under-count |
| 2636015 | New | Programmer | Category C | BUS_ACCESS and BUS_ACCESS_WR PMU events are not reliable |

May 06, 2022: Changes in document version v13.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2457168 | New | Programmer | Category B | AMEVCNTR01 increments incorrectly |
| 2385664 | Updated | Programmer | Category C | Core might not execute any instruction when performing Halting Step |
| 2478655 | New | Programmer | Category C | L2D_CACHE_WB event might over-count |
| 2601282 | New | Programmer | Category C | ELADISABLE does not disable APB access to the complex ELA |

March 02, 2022: Changes in document version v12.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2454944 | New | Programmer | Category A | Unmodified cache line might be written back to memory |
| 2385664 | Updated | Programmer | Category C | Core might not execute any instruction when performing Halting Step |

February 23, 2022: Changes in document version v11.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2420992 | New | Programmer | Category B | Incorrect instructions might be executed |
| 2441009 | New | Programmer | Category B (rare) | Completion of affected memory accesses might not be guaranteed by completion of a TLBI |
| 2226937 | Updated | Programmer | Category C | IMP_CDBGDR0_EL3 read data might be incorrect |
| 2393935 | New | Programmer | Category C | ESR_ELx.EX might be incorrect when stepping a load exclusive |
| 2396657 | New | Programmer | Category C | ESR_ELx.EX might be incorrect due to asynchronous exception when software stepping |
| 2423961 | New | Programmer | Category C | PMU_OVFS event is not generated on PMCCNTR_EL0 overflow |
| 2429106 | New | Programmer | Category C | Exception Catch debug event might not be taken |
| 2429770 | New | Programmer | Category C | PC_WRITE_SPEC event does not count correctly |

January 10, 2022: Changes in document version v10.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2347730 | New | Programmer | Category B | Executing a WFI/WFE with VPU powerdown enabled might result in a deadlock |
| 2371937 | New | Programmer | Category B | Cacheable far atomics might generate data corruption |
| 2358024 | New | Programmer | Category B | EDSCR.INTdis might not mask Non-secure interrupts |
| 2360589 | New | Programmer | Category C | Exception catch might not be taken on ERET from EL3 to any Non-secure EL |
| 2371559 | New | Programmer | Category C | ESR_ELx might be incorrect after trapped vector instruction in Debug state |
| 2385664 | New | Programmer | Category C | Core might not execute any instruction when performing Halting Step |

November 12, 2021: Changes in document version v9.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2008766 | Updated | Programmer | Category B | RAS errors during core power down might cause a deadlock |
| 2164605 | Updated | Programmer | Category C | A watchpoint hit while disabling Halting debug might cause an incorrect debug exception to be taken |
| 2316094 | New | Programmer | Category C | MTE checking might not be reliable |
| 2321712 | New | Programmer | Category C | DLR_EL0[1] is ignored when performing debug exit to A32 |
| 2324165 | New | Programmer | Category C | Processor state might be corrupted if EDSCR.INTdis is set while asynchronous interrupt is in flight |
| 2331844 | New | Programmer | Category C | Software stepping ERET might set PSTATE or SPSR to incorrect value |
| 2335678 | New | Programmer | Category C | BFMMLA/VMMLA result might be corrupted when forwarding source operand from vector load |
| 2341012 | New | Programmer | Category C | Physical SError interrupts while software stepping a load or store instruction might cause two instructions to be stepped |
| 2342004 | New | Programmer | Category C | TLB Parity Check cannot be disabled |
| 2342918 | New | Programmer | Category C | PrefetchTgt transactions are generated when PrefetchTgt is disabled |

September 24, 2021: Changes in document version v8.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2217821 | Updated | Programmer | Category A | Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock |
| 2008766 | Updated | Programmer | Category B | RAS errors during core power down might cause a deadlock |
| 2041909 | Updated | Programmer | Category B | Load-Exclusive/Store-Exclusive loop might not make forward progress |
| 2218134 | New | Programmer | Category B | Single-bit error in branch predictor memory can cause instruction fetch stream corruption |
| 2250311 | New | Programmer | Category B | Asynchronous exceptions while MPMM is active might corrupt processor state |
| 2256538 | New | Programmer | Category B | Hardware breakpoints might not be taken on 32-bit T32 instructions |
| 2288014 | New | Programmer | Category B | Data corruption might occur in rare circumstances |

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2314929 | New | Programmer | Category B | TLB Invalidation prevents core OFF_EMU to OFF transition |
| 2277097 | New | Programmer | Category B (rare) | CMOs to non-shareable locations might deadlock the cluster |
| 2189707 | New | Programmer | Category C | Clean MTE tag writeback might occur after a store to the cache line |
| 2212805 | New | Programmer | Category C | An ECC error while in the active-not-pending state might clear PSTATE.SS |
| 2230537 | New | Programmer | Category C | Load following SVE predicated load/store might cause ordering violation |
| 2236402 | New | Programmer | Category C | Asynchronous exception or debug event might be taken before an exception catch debug event that was generated by an exception entry |
| 2241478 | New | Programmer | Category C | The L1D_CACHE_REFILL and L1D_CACHE_LMISS_RD PMU events might over-count |
| 2266023 | New | Programmer | Category C | Vector stores might not use a faster forwarding path |
| 2265919 | New | Programmer | Category C | Top byte of DLR_EL0 might be incorrect when entering AArch64 halting Debug state |
| 2271084 | New | Programmer | Category C | DTR flags not cleared on external debugger access while leaving Debug state |
| 2272274 | New | Programmer | Category C | Core might execute two instructions on Halting Step debug event |
| 2287488 | New | Programmer | Category C | Core might not execute instruction on Halting Step debug event |
| 2289541 | New | Programmer | Category C | PMU counter overflow can cause spurious PMU_OVFS and PMU_HOVFS events |
| 2289878 | New | Programmer | Category C | Core might step two instructions at once |
| 2291784 | New | Programmer | Category C | PSTATE.IL might be cleared on ERET to Software Stepping |
| 2293834 | New | Programmer | Category C | The core might report that a load-exclusive instruction has not been stepped when it should |
| 2300878 | New | Programmer | Category C | Software stepping ESB might set SPSR.ELx.SS to incorrect value |
| 2303522 | New | Programmer | Category C | Synchronous tag checking on a store exclusive might cause a deadlock if double bit/fatal error seen in L1-Duplicate RAM |

July 21, 2021: Changes in document version v7.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2184257 | New | Programmer | Category A | Normal execution can result in data corruption |
| 2217821 | New | Programmer | Category A | Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock |
| 2064142 | New | Programmer | Category B | A system register write might not take effect after the trace buffer is disabled |
| 2135898 | New | Programmer | Category B | DSPSR_EL0.BTYPE might be incorrect on Debug state entry |
| 2141267 | New | Programmer | Category B | LD1R using SP as base register incorrectly requests MTE Tag Checked |
| 2148497 | New | Programmer | Category B | A trace unit trigger might result in inconsistent register state of the Trace Buffer Extension |
| 2156527 | New | Programmer | Category B | Mapping memory as Tagged when Allocation Tag storage is not provided might lead to CHI protocol violations or Tag corruption |
| 2169012 | New | Programmer | Category B | SnpPreferUnique* might lead to a loss of coherency |
| 2172148 | New | Programmer | Category B | Snp*UniqueFwd might cause a deadlock or data corruption |
| 2218950 | New | Programmer | Category B | Aliased loads and stores might cause an ordering violation |
| 2112025 | New | Programmer | Category C | Instructions might be missed for tracing purposes before Debug state entry due to a watchpoint hit |
| 2120833 | New | Programmer | Category C | DISR_EL1 access in Debug state might be incorrect |
| 2133769 | New | Programmer | Category C | PMU event 0x000C PC_WRITE_RETIRED might be inaccurate for ERET instructions |
| 2135690 | New | Programmer | Category C | PMU events based on STALL_FRONTEND might be inaccurate |
| 2141037 | New | Programmer | Category C | OFF_EMU to OFF power mode transition might result in a deadlock |
| 2146058 | New | Programmer | Category C | PMU events counts might be inaccurate |
| 2161448 | New | Programmer | Category C | PMU_HOVFS event not always exported when self-hosted trace disabled |
| 2164605 | New | Programmer | Category C | A watchpoint hit while disabling Halting debug might cause an incorrect debug exception to be taken |
| 2175229 | New | Programmer | Category C | A single ECC error on the L2 data RAMs might be double-counted |
| 2181318 | New | Programmer | Category C | A PE trace unit trigger might not be reflected in TRBSR_EL1 after a TSB |
| 2187223 | New | Programmer | Category C | In Halting debug state, the core might take an illegal execution exception when it should not |
| 2189260 | New | Programmer | Category C | An MMU fault might not be correctly reflected in the Trace Buffer Extension TRBSR_EL1 register |
| 2217109 | New | Programmer | Category C | RAS error reporting might be incorrect on simultaneous errors |
| 2220074 | New | Programmer | Category C | ETM will indicate that a T32 CC-failed ISB has caused a Context Synchronization Event when it has not |
| 2226937 | New | Programmer | Category C | IMP_CDBGDR0_EL3 read data might be incorrect |

May 06, 2021: Changes in document version v6.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| **1977575** | Updated | Programmer | Category B | Coherency might be lost on a cache line shortly after a core powers up |
| **2022138** | Updated | Programmer | Category B | Core in FULL_RET might not have clock gated |
| **2069954** | New | Programmer | Category B | A hardware update of the dirty bit might occur speculatively if PSTATE.PAN is used |
| **2077057** | New | Programmer | Category B | Software stepping ERETAA or ERETAB might corrupt SPSR_ELx |
| **2082334** | New | Programmer | Category B | ERR2STATUS.SERR might be incorrect |
| **2080326** | New | Programmer | Category B (rare) | A longer TLBI sequence might cause a deadlock |
| **2036369** | New | Programmer | Category C | MPAM3_EL3.TRAPLOWER reset on Cold reset |
| **2053387** | New | Programmer | Category C | Instructions in Debug state after a watchpoint debug event might not be executed |
| **2071968** | New | Programmer | Category C | Incorrect context ID might be associated with trace data |
| **2077160** | New | Programmer | Category C | Incorrect ordering after change in cacheability |
| **2077274** | New | Programmer | Category C | Extra A-sync packet might get written to Trace Buffer in Trace prohibited region |
| **2085871** | New | Programmer | Category C | Shareability aliases might result in incorrect Tag Check Faults |
| **2088637** | New | Programmer | Category C | A watchpointed SVE load might update the FFR register incorrectly |
| **2092740** | New | Programmer | Category C | Debug state exit after execution of a DRPS instruction might lead to deadlock |
| **2096738** | New | Programmer | Category C | Double-bit errors in the duplicate L1 data cache tag RAMs might lead to deadlock |

February 10, 2021: Changes in document version v5.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| **2069773** | New | Programmer | Category A | Forwarding snoops might cause a deadlock |
| **2044951** | New | Programmer | Category B | Entry into FUNC_RET power mode can cause data metastability |
| **2051678** | New | Programmer | Category B | A hardware update of the dirty bit might not be correctly ordered with respect to later instructions |
| **2061513** | New | Programmer | Category B | TRBE might cause a speculative hardware update of the dirty bit outside the trace buffer range |
| **1987184** | New | Programmer | Category C | DBG_RECOV or WARM_RST power modes might lead to deadlock or data corruption |
| **2035302** | New | Programmer | Category C | CONTEXTIDR_EL2 breakpoint matching is enabled at EL3 |
| **2048342** | New | Programmer | Category C | A load or store instruction might hang when encountering multiple uncorrectable or deferred errors |
| **2052374** | New | Programmer | Category C | PMU event ST_RETIRED might be inaccurate for Store-Exclusive instructions |
| **2052205** | New | Programmer | Category C | ERR2STATUS.SERR might be incorrect after an NDErr response is received |
| **2054370** | New | Programmer | Category C | Execution of instructions in Debug state after a watchpoint debug event might incorrectly set EDSCR.ERR and corrupt PSTATE |
| **2056307** | New | Programmer | Category C | A Tag Checked store exclusive might hang if an external abort occurs |
| **2059297** | New | Programmer | Category C | PSTATE.TCO might be unchanged for debug entry due to a watchpoint debug event |

January 22, 2021: Changes in document version v4.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2039708 | New | Programmer | Category A | Non-cacheable stores, maintenance, or speculation restriction instructions might cause a deadlock |
| 2014530 | New | Programmer | Category B | Debug accesses while leaving OFF_EMU power mode might lead to a system deadlock |
| 2015106 | New | Programmer | Category B | Memory mapped accesses to MPMM registers have the wrong offset |
| 2027318 | New | Programmer | Category B | SnpPreferUnique might lead to a loss of coherency |
| 2028010 | New | Programmer | Category B | A PRF{U}M instruction might hang |
| 2038114 | New | Programmer | Category B | EDSCR.STATUS might be incorrect when single stepping a Load-Exclusive instruction |
| 2038923 | New | Programmer | Category B | A change to TRBLIMITR_EL1.E might result in trace buffer corruption |
| 2039165 | New | Programmer | Category B | Debug accesses while leaving OFF_EMU power mode might lead to a system deadlock |
| 2041661 | New | Programmer | Category B | Entry into the Full Retention power mode might cause incorrect execution of a TLB maintenance instruction |
| 2041909 | New | Programmer | Category B | Load-Exclusive/Store-Exclusive loop might not make forward progress |
| 2042739 | New | Programmer | Category B | A store or Load-Exclusive might cause data corruption |
| 1976290 | Updated | Programmer | Category B (rare) | A Tag Checked load might forward incorrect data |
| 2012183 | New | Programmer | Category C | Double-bit errors in the duplicate L1 data cache tag RAMs might lead to deadlock on a snoop |
| 2017921 | New | Programmer | Category C | Exception or DCPS3 instruction in debug state might block execution of instruction from EDITR |
| 2025809 | New | Programmer | Category C | A double-bit error on the L1 data cache MTE tag RAMs might result in a missed or incorrect error record in ERR1STATUS |
| 2033473 | New | Programmer | Category C | A hardware update of the dirty bit might occur speculatively |

December 03, 2020: Changes in document version v3.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1996706 | New | Programmer | Category A | A deadlock might occur during or after WFI or WFE |
| 2007174 | New | Programmer | Category A | Heavy store traffic might lead to a DSB not completing on the same core or another core |
| 1956116 | New | Programmer | Category B | Missing trace for debug entry due to watchpoint hit |
| 1977575 | Updated | Programmer | Category B | Coherency might be lost on a cache line shortly after a core powers up |
| 1981843 | New | Programmer | Category B | A store exclusive to shareable memory other than Normal Inner Write-Back, Outer Write-Back Cacheable might cause a hang |
| 2008766 | New | Programmer | Category B | RAS errors during core power down might cause a deadlock |
| 2022138 | New | Programmer | Category B | Core in FULL_RET might not have clock gated |

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1976290 | New | Programmer | Category B (rare) | A Tag Checked load might forward incorrect data |
| 2002389 | New | Programmer | Category B (rare) | Asynchronous exceptions after ECC errors might cause unpredictable behavior |
| 1925280 | New | Programmer | Category C | A single-bit error in the L2DB RAMs might be reported incorrectly as a corrected error |
| 1934328 | New | Programmer | Category C | Multiple errors detected on the same cycle might lead to an incorrect value of ERR2STATUS and ERR2MISC0 |
| 1946400 | New | Programmer | Category C | Error reporting disabled does not mask error responses on memory accesses |
| 1951568 | Updated | Programmer | Category C | PMU event counts might be inaccurate |
| 1954658 | New | Programmer | Category C | ERR2STATUS.CE might be incorrect |
| 1955046 | New | Programmer | Category C | External events are not observed after Warm reset |
| 1974927 | New | Programmer | Category C | SIMD, FP, and SVE cacheable loads or MTE-checked stores might hang on accessing a poisoned cache line |
| 1975007 | New | Programmer | Category C | WFE trap might take effect when a wake up event is pending |
| 1976399 | New | Programmer | Category C | PrefetchTgt transactions are generated when PrefetchTgt is disabled |
| 1977191 | New | Programmer | Category C | Trigger received immediately out of warm reset might cause deadlock |
| 1980125 | New | Programmer | Category C | A change of ASID without context synchronization might cause memory ordering issues |
| 1980599 | New | Programmer | Category C | An ECC error in the L1 data cache might not be detected |
| 1982956 | New | Programmer | Category C | SIMD and floating point loads to non-gatherable device memory might over-read |
| 1986930 | New | Programmer | Category C | Corruption might occur on MTE allocation tags |
| 1987125 | New | Programmer | Category C | ERR2MISC0.OFR and ERR2MISC0.OFO might not be set on counter overflow |
| 1990749 | New | Programmer | Category C | Errors or poison in the L2 data RAM might lead to deadlock and/or data corruption |
| 1991004 | New | Programmer | Category C | Uncontainable error injection via ERR2PFGCTL might occur at the wrong time |
| 1991342 | New | Programmer | Category C | Traps of System registers in Debug state might cause unexpected exceptions |
| 1996476 | New | Programmer | Category C | A Tag Checked SVE non-fault or first-fault load might hang if an External abort occurs |
| 1996886 | New | Programmer | Category C | ELA connected to warm reset instead of cold reset |
| 1997011 | New | Programmer | Category C | Asynchronous Tag Check fail not synchronized based on SCTLR_ELx.ITFSB for exceptions in Debug state |
| 1998835 | New | Programmer | Category C | Double-bit errors in the duplicate L1 data cache tag RAMs might lead to loss of coherency or deadlock |

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1999032 | New | Programmer | Category C | Non-Data Errors on memory accesses not attributable to a core might not be reported |
| 2000000 | New | Programmer | Category C | Interrupt might not be taken in finite time |
| 1981723 | New | Programmer | Category C | A continuous stream of DVM operations from another PE might block core powerdown |
| 2006188 | New | Programmer | Category C | L2 cache debug operations might hang when no L2 cache is present |
| 2015240 | New | Programmer | Category C | Clearing a pending OS Unlock Catch debug event might cause unpredictable behavior |
| 2016064 | New | Programmer | Category C | A continuous stream of memory requests from one CPU in the complex might block another |

October 23, 2020: Changes in document version v2.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1914417 | New | Programmer | Category A | Coherency may be lost |
| 1921103 | New | Programmer | Category A | TLBI completion might not be guaranteed by DSB |
| 1927171 | New | Programmer | Category A | Mismatched memory attributes might cause deadlock |
| 1943861 | New | Programmer | Category A | Acquire/release semantics violated on store atomic instructions |
| 1946214 | New | Programmer | Category A | Heavy L2 memory traffic may lead to data corruption and deadlock |
| 1968719 | New | Programmer | Category A | Snoops might lead to a deadlock |
| 1970520 | New | Programmer | Category A | Secure Physical Address Cache Invalidate all DVM operation does not invalidate Non-Secure lines in L1 I-Cache |
| 1965350 | New | Programmer | Category A (rare) | Coherency might be lost on a cache line |
| 1975996 | New | Programmer | Category A (rare) | Non-L1-allocating stores might cause data corruption |
| 1910738 | New | Programmer | Category B | Use of tagged memory might cause deadlock, data corruption or incorrect reporting |
| 1922240 | New | Programmer | Category B | SnpPreferUnique* might lead to a loss of coherency |
| 1937669 | New | Programmer | Category B | Tag Checked store might not be correctly ordered by DMB |
| 1942494 | New | Programmer | Category B | A cacheable load of SIMD&FP or SVE registers might return incorrect data |
| 1943339 | New | Programmer | Category B | Cacheable far atomics might result in loss of coherency |
| 1952872 | New | Programmer | Category B | A cacheable load might return incorrect data |
| 1955641 | New | Programmer | Category B | Core in FULL_RET might cause data corruption |
| 1961781 | New | Programmer | Category B | Watchpoints on SVE first-fault and non-fault loads might cause an incorrect value to be loaded and/or an incorrect MTE check results |
| 1962857 | New | Programmer | Category B | Set/way maintenance of L2 cache might cause data corruption |
| 1966377 | New | Programmer | Category B | Incorrect instructions might be executed |
| 1975068 | New | Programmer | Category B | Cacheable Non-shareable mappings might lead to CHI protocol violations |

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1977575 | New | Programmer | Category B | Coherency might be lost on a cache line shortly after a core powers up |
| 1898949 | New | Programmer | Category C | Execution of an atomic instruction may fail to make forward progress |
| 1905896 | New | Programmer | Category C | Error responses to MakeReadUnique transactions might result in data corruption |
| 1911617 | New | Programmer | Category C | Tag Check Fault reporting might be incorrect when ECC error occurs |
| 1915215 | New | Programmer | Category C | Software stepping ERETAA or ERETAB might set SPSR.ELx.SS to incorrect value |
| 1919823 | New | Programmer | Category C | Configuration of IMP_CMPXECTLR_EL1.CDEVC not supported |
| 1921977 | New | Programmer | Category C | Error response to load-exclusive might cause loss of synchronization or deadlock |
| 1904476 | New | Programmer | Category C | Reads of GMID_EL1 might trap to incorrect exception level when MTE is disabled |
| 1924991 | New | Programmer | Category C | Accesses to TRCQCTLR are RAZ/WI instead of UNDEFINED |
| 1929084 | New | Programmer | Category C | Direct access to L1 data cache MTE data RAMs does not function |
| 1933869 | New | Programmer | Category C | Writes to DBGWCR<n>_EL1/DBGWVR<n>_EL1 might cause speculative reads of device memory |
| 1951345 | New | Programmer | Category C | PMU_HOVFS event exported when EL2 trace disabled |
| 1951423 | New | Programmer | Category C | An ECC error during core power down might cause another error to occur after power up |
| 1951568 | New | Programmer | Category C | PMU event counts might be inaccurate |
| 1954191 | New | Programmer | Category C | L2 cache debug operations might hang and block a power off request |
| 1955072 | New | Programmer | Category C | Mapping memory as Tagged when Allocation Tag storage is not provided might lead to CHI protocol violations |
| 1956538 | New | Programmer | Category C | Execution of ESB instruction in debug state might cause unpredictable behavior |
| 1959615 | New | Programmer | Category C | Exceptions in debug state might allow execution of subsequent instruction in EDITR |
| 1964078 | New | Programmer | Category C | VMID tracing incorrectly allowed or disallowed based on TRFCR_EL2.CX |
| 1965154 | New | Programmer | Category C | PMU snapshot records incorrect Context ID when PMU inactive |
| 1966395 | New | Programmer | Category C | A continuous stream of stores might prevent forward progress of a coherency operation |

July 17, 2020: Changes in document version v1.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1902691 | New | Programmer | Category B | Trace Buffer Extension can cause trace corruption or deadlock |

# Errata summary table

The errata associated with this product affect the product versions described in the following table.

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 1914417 | Programmer | Category A | Coherency may be lost | r0p0 | r0p1 |
| 1921103 | Programmer | Category A | TLBI completion might not be guaranteed by DSB | r0p0 | r0p1 |
| 1927171 | Programmer | Category A | Mismatched memory attributes might cause deadlock | r0p0 | r0p1 |
| 1943861 | Programmer | Category A | Acquire/release semantics violated on store atomic instructions | r0p0 | r0p1 |
| 1946214 | Programmer | Category A | Heavy L2 memory traffic may lead to data corruption and deadlock | r0p0 | r0p1 |
| 1968719 | Programmer | Category A | Snoops might lead to a deadlock | r0p0 | r0p1 |
| 1970520 | Programmer | Category A | Secure Physical Address Cache Invalidate all DVM operation does not invalidate Non-Secure lines in L1 I-Cache | r0p0 | r0p1 |
| 1996706 | Programmer | Category A | A deadlock might occur during or after WFI or WFE | r0p0 | r0p1 |
| 2007174 | Programmer | Category A | Heavy store traffic might lead to a DSB not completing on the same core or another core | r0p0 | r0p1 |
| 2039708 | Programmer | Category A | Non-cacheable stores, maintenance, or speculation restriction instructions might cause a deadlock | r0p0, r0p1 | r0p2 |
| 2069773 | Programmer | Category A | Forwarding snoops might cause a deadlock | r0p0, r0p1 | r0p2 |
| 2184257 | Programmer | Category A | Normal execution can result in data corruption | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2217821 | Programmer | Category A | Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2454944 | Programmer | Category A | Unmodified cache line might be written back to memory | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 1965350 | Programmer | Category A (rare) | Coherency might be lost on a cache line | r0p0 | r0p1 |
| 1975996 | Programmer | Category A (rare) | Non-L1-allocating stores might cause data corruption | r0p0, r0p1 | r0p2 |
| 1902691 | Programmer | Category B | Trace Buffer Extension can cause trace corruption or deadlock | r0p0, r0p1 | r0p2 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 1910738 | Programmer | Category B | Use of tagged memory might cause deadlock, data corruption or incorrect reporting | r0p0 | r0p1 |
| 1922240 | Programmer | Category B | SnpPreferUnique* might lead to a loss of coherency | r0p0 | r0p1 |
| 1937669 | Programmer | Category B | Tag Checked store might not be correctly ordered by DMB | r0p0 | r0p1 |
| 1942494 | Programmer | Category B | A cacheable load of SIMD&FP or SVE registers might return incorrect data | r0p0 | r0p1 |
| 1943339 | Programmer | Category B | Cacheable far atomics might result in loss of coherency | r0p0 | r0p1 |
| 1952872 | Programmer | Category B | A cacheable load might return incorrect data | r0p0 | r0p1 |
| 1955641 | Programmer | Category B | Core in FULL_RET might cause data corruption | r0p0 | r0p1 |
| 1956116 | Programmer | Category B | Missing trace for debug entry due to watchpoint hit | r0p0 | r0p1 |
| 1961781 | Programmer | Category B | Watchpoints on SVE first-fault and non-fault loads might cause an incorrect value to be loaded and/or an incorrect MTE check results | r0p0 | r0p1 |
| 1962857 | Programmer | Category B | Set/way maintenance of L2 cache might cause data corruption | r0p0 | r0p1 |
| 1966377 | Programmer | Category B | Incorrect instructions might be executed | r0p0 | r0p1 |
| 1975068 | Programmer | Category B | Cacheable Non-shareable mappings might lead to CHI protocol violations | r0p0 | r0p1 |
| 1977575 | Programmer | Category B | Coherency might be lost on a cache line shortly after a core powers up | r0p0 | r0p1 |
| 1981843 | Programmer | Category B | A store exclusive to shareable memory other than Normal Inner Write-Back, Outer Write-Back Cacheable might cause a hang | r0p0 | r0p1 |
| 2008766 | Programmer | Category B | RAS errors during core power down might cause a deadlock | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2014530 | Programmer | Category B | Debug accesses while leaving OFF_EMU power mode might lead to a system deadlock | r0p0, r0p1 | r0p2 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| **2015106** | Programmer | Category B | Memory mapped accesses to MPMM registers have the wrong offset | r0p0 | r0p1 |
| **2022138** | Programmer | Category B | Core in FULL_RET might not have clock gated | r0p0, r0p1 | r0p2 |
| **2027318** | Programmer | Category B | SnpPreferUnique might lead to a loss of coherency | r0p0, r0p1 | r0p2 |
| **2028010** | Programmer | Category B | A PRF{U}M instruction might hang | r0p0, r0p1 | r0p2 |
| **2038114** | Programmer | Category B | EDSCR.STATUS might be incorrect when single stepping a Load-Exclusive instruction | r0p0, r0p1, r0p2 | r0p3 |
| **2038923** | Programmer | Category B | A change to TRBLIMITR_EL1.E might result in trace buffer corruption | r0p0, r0p1, r0p2 | r0p3 |
| **2039165** | Programmer | Category B | Debug accesses while leaving OFF_EMU power mode might lead to a system deadlock | r0p0, r0p1, r0p2 | r0p3 |
| **2041661** | Programmer | Category B | Entry into the Full Retention power mode might cause incorrect execution of a TLB maintenance instruction | r0p0, r0p1 | r0p2 |
| **2041909** | Programmer | Category B | Load-Exclusive/Store-Exclusive loop might not make forward progress | r0p0, r0p1, r0p2 | r0p3 |
| **2042739** | Programmer | Category B | A store or Load-Exclusive might cause data corruption | r0p0, r0p1, r0p2 | r0p3 |
| **2044951** | Programmer | Category B | Entry into FUNC_RET power mode can cause data metastability | r0p0, r0p1, r0p2 | r0p3 |
| **2051678** | Programmer | Category B | A hardware update of the dirty bit might not be correctly ordered with respect to later instructions | r0p0, r0p1, r0p2 | r0p3 |
| **2061513** | Programmer | Category B | TRBE might cause a speculative hardware update of the dirty bit outside the trace buffer range | r0p0, r0p1, r0p2 | r0p3 |
| **2064142** | Programmer | Category B | A system register write might not take effect after the trace buffer is disabled | r0p0, r0p1, r0p2 | r0p3 |
| **2069954** | Programmer | Category B | A hardware update of the dirty bit might occur speculatively if PSTATE.PAN is used | r0p0, r0p1, r0p2 | r0p3 |
| **2077057** | Programmer | Category B | Software stepping ERETAA or ERETAB might corrupt SPSR_ELx | r0p0, r0p1, r0p2 | r0p3 |
| **2082334** | Programmer | Category B | ERR2STATUS.SERR might be incorrect | r0p0, r0p1, r0p2, r0p3 | r1p0 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2135898 | Programmer | Category B | DSPSR_EL0.BTYPE might be incorrect on Debug state entry | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2141267 | Programmer | Category B | LD1R using SP as base register incorrectly requests MTE Tag Checked | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2148497 | Programmer | Category B | A trace unit trigger might result in inconsistent register state of the Trace Buffer Extension | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2156527 | Programmer | Category B | Mapping memory as Tagged when Allocation Tag storage is not provided might lead to CHI protocol violations or Tag corruption | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2169012 | Programmer | Category B | SnpPreferUnique* might lead to a loss of coherency | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2172148 | Programmer | Category B | Snp*UniqueFwd might cause a deadlock or data corruption | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2218134 | Programmer | Category B | Single-bit error in branch predictor memory can cause instruction fetch stream corruption | r1p0 | r1p1 |
| 2218950 | Programmer | Category B | Aliased loads and stores might cause an ordering violation | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2250311 | Programmer | Category B | Asynchronous exceptions while MPMM is active might corrupt processor state | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2256538 | Programmer | Category B | Hardware breakpoints might not be taken on 32-bit T32 instructions | r1p0 | r1p1 |
| 2288014 | Programmer | Category B | Data corruption might occur in rare circumstances | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2314929 | Programmer | Category B | TLB Invalidation prevents core OFF_EMU to OFF transition | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2347730 | Programmer | Category B | Executing a WFI/WFE with VPU powerdown enabled might result in a deadlock | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2358024 | Programmer | Category B | EDSCR.INTdis might not mask Non-secure interrupts | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2371937 | Programmer | Category B | Cacheable far atomics might generate data corruption | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2420992 | Programmer | Category B | Incorrect instructions might be executed | r1p0, r1p1 | r1p2 |
| 2457168 | Programmer | Category B | AMEVCNTR01 increments incorrectly | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2658417 | Programmer | Category B | BFMMLA or VMMLA instructions might produce incorrect result | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| **2666669** | Programmer | Category B | Data corruption might occur during core powerdown | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| **2675636** | Programmer | Category B | An asynchronous MTE check might not observe correct memory ordering | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| **2684597** | Programmer | Category B | A core might deadlock during powerdown if TRBE is enabled | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| **2724177** | Programmer | Category B | Deferred error might become uncontainable | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| **2971420** | Programmer | Category B | Data corruption or deadlock might happen if TRBE is enabled | r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| **3057514** | Programmer | Category B | Deferred error might become uncontainable when BROADCASTMTE is set | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| **3117295** | Programmer | Category B | A speculatively executed unprivileged load might leak data from a privileged via side channel | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| **1976290** | Programmer | Category B (rare) | A Tag Checked load might forward incorrect data | r0p0 | r0p1 |
| **2002389** | Programmer | Category B (rare) | Asynchronous exceptions after ECC errors might cause unpredictable behavior | r0p0, r0p1 | r0p2 |
| **2080326** | Programmer | Category B (rare) | A longer TLBI sequence might cause a deadlock | r0p0, r0p1, r0p2 | r0p3 |
| **2277097** | Programmer | Category B (rare) | CMOs to non-shareable locations might deadlock the cluster | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| **2441009** | Programmer | Category B (rare) | Completion of affected memory accesses might not be guaranteed by completion of a TLBI | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| **1898949** | Programmer | Category C | Execution of an atomic instruction may fail to make forward progress | r0p0 | r0p1 |
| **1904476** | Programmer | Category C | Reads of GMID_EL1 might trap to incorrect exception level when MTE is disabled | r0p0 | r0p1 |
| **1905896** | Programmer | Category C | Error responses to MakeReadUnique transactions might result in data corruption | r0p0 | r0p1 |
| **1911617** | Programmer | Category C | Tag Check Fault reporting might be incorrect when ECC error occurs | r0p0 | r0p1 |
| **1915215** | Programmer | Category C | Software stepping ERETAA or ERETAB might set SPSR.ELx.SS to incorrect value | r0p0 | r0p1 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 1919823 | Programmer | Category C | Configuration of IMP_CMPXECTLR_EL1.CDEVC not supported | r0p0 | r0p1 |
| 1921977 | Programmer | Category C | Error response to load-exclusive might cause loss of synchronization or deadlock | r0p0 | r0p1 |
| 1924991 | Programmer | Category C | Accesses to TRCQCTLR are RAZ/WI instead of UNDEFINED | r0p0 | r0p1 |
| 1925280 | Programmer | Category C | A single-bit error in the L2DB RAMs might be reported incorrectly as a corrected error | r0p0 | r0p1 |
| 1929084 | Programmer | Category C | Direct access to L1 data cache MTE data RAMs does not function | r0p0 | r0p1 |
| 1933869 | Programmer | Category C | Writes to DBGWCR<n>_EL1/DBGWVR<n>_EL1 might cause speculative reads of device memory | r0p0 | r0p1 |
| 1934328 | Programmer | Category C | Multiple errors detected on the same cycle might lead to an incorrect value of ERR2STATUS and ERR2MISC0 | r0p0, r0p1 | r0p2 |
| 1946400 | Programmer | Category C | Error reporting disabled does not mask error responses on memory accesses | r0p0 | r0p1 |
| 1951345 | Programmer | Category C | PMU_HOVFS event exported when EL2 trace disabled | r0p0 | r0p1 |
| 1951423 | Programmer | Category C | An ECC error during core power down might cause another error to occur after power up | r0p0 | r0p1 |
| 1951568 | Programmer | Category C | PMU event counts might be inaccurate | r0p0 | r0p1 |
| 1954191 | Programmer | Category C | L2 cache debug operations might hang and block a power off request | r0p0 | r0p1 |
| 1954658 | Programmer | Category C | ERR2STATUS.CE might be incorrect | r0p0 | r0p1 |
| 1955046 | Programmer | Category C | External events are not observed after Warm reset | r0p0 | r0p1 |
| 1955072 | Programmer | Category C | Mapping memory as Tagged when Allocation Tag storage is not provided might lead to CHI protocol violations | r0p0 | r0p1 |
| 1956538 | Programmer | Category C | Execution of ESB instruction in debug state might cause unpredictable behavior | r0p0 | r0p1 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| **1959615** | Programmer | Category C | Exceptions in debug state might allow execution of subsequent instruction in EDITR | r0p0 | r0p1 |
| **1964078** | Programmer | Category C | VMID tracing incorrectly allowed or disallowed based on TRFCR_EL2.CX | r0p0 | r0p1 |
| **1965154** | Programmer | Category C | PMU snapshot records incorrect Context ID when PMU inactive | r0p0 | r0p1 |
| **1966395** | Programmer | Category C | A continuous stream of stores might prevent forward progress of a coherency operation | r0p0 | r0p1 |
| **1974927** | Programmer | Category C | SIMD, FP, and SVE cacheable loads or MTE-checked stores might hang on accessing a poisoned cache line | r0p0 | r0p1 |
| **1975007** | Programmer | Category C | WFE trap might take effect when a wake up event is pending | r0p0 | r0p1 |
| **1976399** | Programmer | Category C | PrefetchTgt transactions are generated when PrefetchTgt is disabled | r0p0 | r0p1 |
| **1977191** | Programmer | Category C | Trigger received immediately out of warm reset might cause deadlock | r0p0 | r0p1 |
| **1980125** | Programmer | Category C | A change of ASID without context synchronization might cause memory ordering issues | r0p0 | r0p1 |
| **1980599** | Programmer | Category C | An ECC error in the L1 data cache might not be detected | r0p0 | r0p1 |
| **1981723** | Programmer | Category C | A continuous stream of DVM operations from another PE might block core powerdown | r0p0 | r0p1 |
| **1982956** | Programmer | Category C | SIMD and floating point loads to non-gatherable device memory might over-read | r0p0 | r0p1 |
| **1986930** | Programmer | Category C | Corruption might occur on MTE allocation tags | r0p0 | r0p1 |
| **1987125** | Programmer | Category C | ERR2MISC0.OFR and ERR2MISC0.OFO might not be set on counter overflow | r0p0 | r0p1 |
| **1987184** | Programmer | Category C | DBG_RECOV or WARM_RST power modes might lead to deadlock or data corruption | r0p0, r0p1 | r0p2 |
| **1990749** | Programmer | Category C | Errors or poison in the L2 data RAM might lead to deadlock and/or data corruption | r0p0 | r0p1 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 1991004 | Programmer | Category C | Uncontainable error injection via ERR2PFGCTL might occur at the wrong time | r0p0 | r0p1 |
| 1991342 | Programmer | Category C | Traps of System registers in Debug state might cause unexpected exceptions | r0p0 | r0p1 |
| 1996476 | Programmer | Category C | A Tag Checked SVE non-fault or first-fault load might hang if an External abort occurs | r0p0 | r0p1 |
| 1996886 | Programmer | Category C | ELA connected to warm reset instead of cold reset | r0p0 | r0p1 |
| 1997011 | Programmer | Category C | Asynchronous Tag Check fail not synchronized based on SCTLR_ELx.ITFSB for exceptions in Debug state | r0p0 | r0p1 |
| 1998835 | Programmer | Category C | Double-bit errors in the duplicate L1 data cache tag RAMs might lead to loss of coherency or deadlock | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 1999032 | Programmer | Category C | Non-Data Errors on memory accesses not attributable to a core might not be reported | r0p0 | r0p1 |
| 2000000 | Programmer | Category C | Interrupt might not be taken in finite time | r0p0 | r0p1 |
| 2006188 | Programmer | Category C | L2 cache debug operations might hang when no L2 cache is present | r0p0, r0p1 | r0p2 |
| 2012183 | Programmer | Category C | Double-bit errors in the duplicate L1 data cache tag RAMs might lead to deadlock on a snoop | r0p0, r0p1 | r0p2 |
| 2015240 | Programmer | Category C | Clearing a pending OS Unlock Catch debug event might cause unpredictable behavior | r0p0 | r0p1 |
| 2016064 | Programmer | Category C | A continuous stream of memory requests from one CPU in the complex might block another | r0p0 | r0p1 |
| 2017921 | Programmer | Category C | Exception or DCPS3 instruction in debug state might block execution of instruction from EDITR | r0p0, r0p1 | r0p2 |
| 2025809 | Programmer | Category C | A double-bit error on the L1 data cache MTE tag RAMs might result in a missed or incorrect error record in ERR1STATUS | r0p0, r0p1 | r0p2 |
| 2033473 | Programmer | Category C | A hardware update of the dirty bit might occur speculatively | r0p0, r0p1 | r0p2 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2035302 | Programmer | Category C | CONTEXTIDR_EL2 breakpoint matching is enabled at EL3 | r0p0, r0p1, r0p2 | r0p3 |
| 2036369 | Programmer | Category C | MPAM3_EL3.TRAPLOWER reset on Cold reset | r0p0, r0p1 | r0p2 |
| 2048342 | Programmer | Category C | A load or store instruction might hang when encountering multiple uncorrectable or deferred errors | r0p0, r0p1, r0p2 | r0p3 |
| 2052205 | Programmer | Category C | ERR2STATUS.SERR might be incorrect after an NDErr response is received | r0p0, r0p1, r0p2 | r0p3 |
| 2052374 | Programmer | Category C | PMU event ST_RETIRED might be inaccurate for Store-Exclusive instructions | r0p0, r0p1, r0p2 | r0p3 |
| 2053387 | Programmer | Category C | Instructions in Debug state after a watchpoint debug event might not be executed | r0p0, r0p1, r0p2 | r0p3 |
| 2054370 | Programmer | Category C | Execution of instructions in Debug state after a watchpoint debug event might incorrectly set EDSCR.ERR and corrupt PSTATE | r0p0, r0p1, r0p2 | r0p3 |
| 2056307 | Programmer | Category C | A Tag Checked store exclusive might hang if an external abort occurs | r0p0, r0p1, r0p2 | r0p3 |
| 2059297 | Programmer | Category C | PSTATE.TCO might be unchanged for debug entry due to a watchpoint debug event | r0p0, r0p1, r0p2 | r0p3 |
| 2071968 | Programmer | Category C | Incorrect context ID might be associated with trace data | r0p0, r0p1, r0p2 | r0p3 |
| 2077160 | Programmer | Category C | Incorrect ordering after change in cacheability | r0p0, r0p1, r0p2 | r0p3 |
| 2077274 | Programmer | Category C | Extra A-sync packet might get written to Trace Buffer in Trace prohibited region | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2085871 | Programmer | Category C | Shareability aliases might result in incorrect Tag Check Faults | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2088637 | Programmer | Category C | A watchpointed SVE load might update the FFR register incorrectly | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2092740 | Programmer | Category C | Debug state exit after execution of a DRPS instruction might lead to deadlock | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2096738 | Programmer | Category C | Double-bit errors in the duplicate L1 data cache tag RAMs might lead to deadlock | r0p0, r0p1, r0p2, r0p3 | r1p0 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2112025 | Programmer | Category C | Instructions might be missed for tracing purposes before Debug state entry due to a watchpoint hit | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2120833 | Programmer | Category C | DISR_EL1 access in Debug state might be incorrect | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2133769 | Programmer | Category C | PMU event 0x000C PC_WRITE_RETIRED might be inaccurate for ERET instructions | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2135690 | Programmer | Category C | PMU events based on STALL_FRONTEND might be inaccurate | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2141037 | Programmer | Category C | OFF_EMU to OFF power mode transition might result in a deadlock | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2146058 | Programmer | Category C | PMU events counts might be inaccurate | r0p0, r0p1, r0p2, r0p3 | r1p0 |
| 2161448 | Programmer | Category C | PMU_HOVFS event not always exported when self-hosted trace disabled | r0p1, r0p2, r0p3 | r1p0 |
| 2164605 | Programmer | Category C | A watchpoint hit while disabling Halting debug might cause an incorrect debug exception to be taken | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2175229 | Programmer | Category C | A single ECC error on the L2 data RAMs might be double-counted | r1p0 | r1p1 |
| 2181318 | Programmer | Category C | A PE trace unit trigger might not be reflected in TRBSR_EL1 after a TSB | r1p0 | r1p1 |
| 2187223 | Programmer | Category C | In Halting debug state, the core might take an illegal execution exception when it should not | r1p0 | r1p1 |
| 2189260 | Programmer | Category C | An MMU fault might not be correctly reflected in the Trace Buffer Extension TRBSR_EL1 register | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2189707 | Programmer | Category C | Clean MTE tag writeback might occur after a store to the cache line | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2212805 | Programmer | Category C | An ECC error while in the active-not-pending state might clear PSTATE.SS | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2217109 | Programmer | Category C | RAS error reporting might be incorrect on simultaneous errors | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2220074 | Programmer | Category C | ETM will indicate that a T32 CC-failed ISB has caused a Context Synchronization Event when it has not | r1p0 | r1p1 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2226937 | Programmer | Category C | IMP_CDBGDR0_EL3 read data might be incorrect | r1p0 | r1p1 |
| 2230537 | Programmer | Category C | Load following SVE predicated load/store might cause ordering violation | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2236402 | Programmer | Category C | Asynchronous exception or debug event might be taken before an exception catch debug event that was generated by an exception entry | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2241478 | Programmer | Category C | The L1D_CACHE_REFILL and L1D_CACHE_LMISS_RD PMU events might over-count | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2265919 | Programmer | Category C | Top byte of DLR_EL0 might be incorrect when entering AArch64 halting Debug state | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2266023 | Programmer | Category C | Vector stores might not use a faster forwarding path | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2271084 | Programmer | Category C | DTR flags not cleared on external debugger access while leaving Debug state | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2272274 | Programmer | Category C | Core might execute two instructions on Halting Step debug event | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2287488 | Programmer | Category C | Core might not execute instruction on Halting Step debug event | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2289541 | Programmer | Category C | PMU counter overflow can cause spurious PMU_OVFS and PMU_HOVFS events | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2289878 | Programmer | Category C | Core might step two instructions at once | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2291784 | Programmer | Category C | PSTATE.IL might be cleared on ERET to Software Stepping | r0p0, r0p1, r0p2, r0p3, r1p0 | r1p1 |
| 2293834 | Programmer | Category C | The core might report that a load-exclusive instruction has not been stepped when it should | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2300878 | Programmer | Category C | Software stepping ESB might set SPSR.ELx.SS to incorrect value | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2303522 | Programmer | Category C | Synchronous tag checking on a store exclusive might cause a deadlock if double bit/fatal error seen in L1-Duplicate RAM | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2316094 | Programmer | Category C | MTE checking might not be reliable | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2321712 | Programmer | Category C | DLR_EL0[1] is ignored when performing debug exit to A32 | r1p0, r1p1 | r1p2 |
| 2324165 | Programmer | Category C | Processor state might be corrupted if EDSCR.INTdis is set while asynchronous interrupt is in flight | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2331844 | Programmer | Category C | Software stepping ERET might set PSTATE or SPSR to incorrect value | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2335678 | Programmer | Category C | BFMMLA/VMMLA result might be corrupted when forwarding source operand from vector load | r1p0, r1p1 | r1p2 |
| 2341012 | Programmer | Category C | Physical SError interrupts while software stepping a load or store instruction might cause two instructions to be stepped | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2342004 | Programmer | Category C | TLB Parity Check cannot be disabled | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2342918 | Programmer | Category C | PrefetchTgt transactions are generated when PrefetchTgt is disabled | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2360589 | Programmer | Category C | Exception catch might not be taken on ERET from EL3 to any Non-secure EL | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2371559 | Programmer | Category C | ESR_ELx might be incorrect after trapped vector instruction in Debug state | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2385664 | Programmer | Category C | Core might not execute any instruction when performing Halting Step | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2393935 | Programmer | Category C | ESR_ELx.EX might be incorrect when stepping a load exclusive | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2396657 | Programmer | Category C | ESR_ELx.EX might be incorrect due to asynchronous exception when software stepping | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2423961 | Programmer | Category C | PMU_OVFS event is not generated on PMCCNTR_EL0 overflow | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2429106 | Programmer | Category C | Exception Catch debug event might not be taken | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2429770 | Programmer | Category C | PC_WRITE_SPEC event does not count correctly | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2478655 | Programmer | Category C | L2D_CACHE_WB event might over-count | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2601282 | Programmer | Category C | ELADISABLE does not disable APB access to the complex ELA | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2618004 | Programmer | Category C | LDST_SPEC event might under-count | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2636015 | Programmer | Category C | BUS_ACCESS and BUS_ACCESS_WR PMU events are not reliable | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 | r1p2 |
| 2679270 | Programmer | Category C | External aborts reporting can not be disabled | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2688792 | Programmer | Category C | PMU event L3D_CACHE_LMISS_RD might be inaccurate | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2690487 | Programmer | Category C | Some architectural PMU events are not always available to trace unit | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2710402 | Programmer | Category C | Read value of IMP_CPUCFR_EL1 might be incorrect | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2713359 | Programmer | Category C | ERR2STATUS.UET field might be incorrect | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2718749 | Programmer | Category C | Cache debug target for L2 Data RAM may not record correct data | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2736240 | Programmer | Category C | ERR2STATUS might be incorrect | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 | r1p3 |
| 2834345 | Programmer | Category C | Direct access to internal memory might not be reliable | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2856823 | Programmer | Category C | Speculative dirty bit hardware update might happen for store operation | r1p0, r1p1, r1p2, r1p3 | Open |
| 2867016 | Programmer | Category C | An uncontainable error might deadlock the cluster | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2867018 | Programmer | Category C | Error record registers indicate incorrect feature support in configurations without cache protection | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2878694 | Programmer | Category C | LDG or MTE checked load/store might fail to detect poisoned data | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2879976 | Programmer | Category C | Unmodified cache line might be written back to memory | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2971621 | Programmer | Category C | Unmodified pagetable cachelines might be written back to memory | r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3 | Open |
| 2976328 | Programmer | Category C | Store data might be lost when a correctable error is detected in the L1 data cache | r1p1, r1p2, r1p3 | Open |

# Errata descriptions

## Category A

### 1914417
### Coherency may be lost

#### Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain conditions, coherency may be lost on a cache line, resulting in potential data corruption.

#### Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache, by setting the L2_CACHE parameter to TRUE.

#### Conditions

1. Data caching is enabled.
2. The MMU is enabled.
3. Unlikely micro-architectural conditions occur.

#### Implications

If the conditions above are met, coherency may be lost for lines in the caches, resulting in possible data corruption.

#### Workaround

There is no workaround for this erratum.

# 1921103
# TLBI completion might not be guaranteed by DSB

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A TLB invalidate (**`TLBI`**) followed by a **`DSB`** on any PE in the system might not guarantee all memory accesses performed by a core using the TLB entries that have been invalidated by the previous **`TLBI`** operation are complete.

## Configurations Affected

All configurations are affected.

## Conditions

1. A Cortex-A510 core starts a memory access.
2. A different PE in the system executes a TLB invalidate followed by a **`DSB`**. This **`TLBI`** invalidates the TLB entry used by the memory access in condition 1.

## Implications

A memory access using a TLB entry that has been invalidated by the subsequent **`TLBI`**, might not have been completed when the **`DSB`** following the **`TLBI`** finishes.

## Workaround

This erratum can be avoided if the **`TLBI`** and **`DSB`** sequence is repeated. After the completion of the second **`DSB`**, the stores that should have been completed before the first **`DSB`** will complete. This workaround might impact system performance significantly.

# 1927171
# Mismatched memory attributes might cause deadlock

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If mismatched memory attributes are used, the system might deadlock.

## Configurations Affected

Configurations with two cores per complex are affected.

## Conditions

1. Both cores in the complex have a cache line, A, in the L1 data cache, with cacheable attributes.
2. One of the cores issues a non-cacheable transaction to the same physical address as cache line A.
3. Unlikely microarchitectural conditions involving the exact timing of multiple operations occur.

## Implications

If the above conditions are met, then one or more cores or other coherent agents in the system might deadlock.

If the cluster does not receive external snoops to addresses also mapped as non-cacheable, the probability of hitting this erratum is significantly reduced.

## Workaround

There is no workaround.

# 1943861 Acquire/release semantics violated on store atomic instructions

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Store atomic instructions with release semantics do not guarantee RCsc (Release Consistency sequentially consistent) consistency with respect to a following Load-Acquire.

## Configurations Affected

All configurations are affected.

## Conditions

A core executes the following sequence of instructions with no LD+ST barrier (**DMB**/**DSB ISH**/**OSH**/**NSH**/**SY**) in between:

1. A store atomic instruction with release semantics (**STADDL**, **STCLRL**, **STEORL**, **STSETL**, **STSMAXL**, **STSMINL**, **STUMAXL**, **STUMINL**)
2. A load with Load-Acquire RCsc semantics

## Implications

If the conditions are met, the Load-Acquire does not guarantee the Store-Release is observed by other PEs before the memory access generated by the Load-Acquire.

## Workaround

There is no workaround.

# 1946214
# Heavy L2 memory traffic may lead to data corruption and deadlock

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Heavy memory traffic in the L2 cache may lead to data corruption and a subsequent deadlock.

## Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache, and with two L2 RAM partitions per slice (configuration parameter L2_CACHE is TRUE and L2_NUM_RAMCTL_PARTITIONS is 2).

## Conditions

1. Heavy memory traffic in the L2 cache.
2. Complex microarchitectural conditions occur.

## Implications

If the conditions are met, data corruption may occur, followed by a deadlock.

## Workaround

There is no workaround.

## 1968719
## Snoops might lead to a deadlock

### Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

Snoops received by a complex, either from other coherent requestors in the cluster or external coherency requests, might lead to a deadlock if the snooped cache line is currently being written back or evicted by the complex.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. The complex performs a cache writeback, either because of a natural eviction, or a non-allocating write.
2. The cache writeback request cannot be sent immediately due to structural backpressure on the request channel.
3. The complex receives a snoop to the same address as the cache writeback, either from other coherent requestors in the cluster or external coherency requests.
4. A downstream memory component, such as the DSU, has a dependency, structural or otherwise, that keeps structural backpressure on the request channel until the snoop completes.

### Implications

If the above conditions are met, the complex might be unable to respond to the snoop, leading to a deadlock.

### Workaround

There is no workaround.

# 1970520
# Secure Physical Address Cache Invalidate all DVM operation does not invalidate Non-Secure lines in L1 I-Cache

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

On receipt of a Secure Physical Address Cache Invalidate all DVM operation, the core will invalidate all Secure lines from the L1 I-cache, but not Non-Secure lines. Due to an AMBA architecture erratum, this is no longer legal behavior.

## Configurations Affected

This erratum affects configurations where Cortex-A510 is used with cores that can generate Secure Physical Instruction Cache Invalidate all DVM operations. Cortex-A510 will not generate this kind of DVM operation.

## Conditions

1. The core receives a Secure Physical Address Cache Invalidate all DVM operation. These operations will be generated by a core executing an `IC IALLU` or `IC IALLUIS` instruction. Cortex-A510 cores will not generate this type of operation.

## Implications

Non-secure software might observe stale instruction cache contents after an invalidation from Secure software.

## Workaround

There is no workaround.

## 1996706
## A deadlock might occur during or after WFI or WFE

### Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If a core's L1 data cache is snooped while it is in WFI or WFE low-power state, the core might later deadlock on a coherency operation or write.

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under the following conditions:

1. A core executes a `WFI` or `WFE` instruction. As soon as outstanding traffic is drained, this results in the core clock being gated.
2. One or more cache lines are still present in the L1 data cache at this time.
3. Another core in the same shareability domain causes a coherency request to a cache line present in the L1 data cache of the core executing the `WFE` or `WFI` instruction above.

### Implications

If these conditions are met, the core or a coherency operation to the core might deadlock at the time the core is in WFI or WFE low-power state, or at a later time.

### Workaround

There is no workaround.

## 2007174
## Heavy store traffic might lead to a DSB not completing on the same core or another core

### Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

A store might lead to a `DSB` on any core in the same shareability domain to hang if one of the cores in the complex has generated heavy store or cache maintenance traffic.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. One or more cores in the complex cause heavy store or cache maintenance traffic beyond the L1 cache.
2. A core in the complex executes a store.
3. Very unusual, timing-sensitive, microarchitectural conditions occur.

### Implications

If these conditions are met, a `DSB` on any core in the same shareability domain during or after the stores might hang.

### Workaround

There is no workaround.

## 2039708
## Non-cacheable stores, maintenance, or speculation restriction instructions might cause a deadlock

### Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

### Description

A deadlock might occur if both cores are simultaneously performing a sequence of stores, TLB or instruction cache maintenance instructions, or speculation restriction instructions.

### Configurations Affected

This erratum affects configurations with two cores in a complex.

### Conditions

The erratum occurs under the following conditions:

1. Both cores in the complex simultaneously perform a sequence of 9 or more of one or more of the following instructions without barriers in between the accesses:
   - Stores to memory other than Normal Inner-Writeback and Outer-Writeback Cacheable
   - TLB maintenance instructions (`TLBI`)
   - Instruction cache maintenance instructions (`IC`)
   - Speculation restriction instructions (`CPP`, `DVP`, `CFP`)
2. Unlikely, timing-sensitive microarchitectural conditions occur.

### Implications

If the conditions are met, a deadlock affecting all cores in the same shareability domain might occur. Arm believes this is unlikely to cause a problem for sample silicon. Please contact Arm if you believe you are encountering this issue for support with software mitigations.

### Workaround

There is no workaround.

# 2069773
# Forwarding snoops might cause a deadlock

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

If the complex receives a forwarding snoop as part of a coherency request from another core or requester in the system, a deadlock might occur.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The complex receives a forwarding snoop as part of a coherency request from another core or requester in the system.
2. Unlikely, but not rare, timing-sensitive microarchitectural conditions occur.

## Implications

If the conditions are met, a deadlock might occur.

## Workaround

There is no workaround.

# 2184257
# Normal execution can result in data corruption

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

A sequence of loads and stores might result in data that is forwarded to dependent instructions to be inconsistent with data that is written to the architectural register file.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. A store to a memory location A is executed.
2. A load to the same 64-byte cache line as memory location A occurs, but not overlapping with A.
3. A load to the same 64-byte cache line as memory location A occurs.
4. A concurrent store to memory location A is executed on another PE. The store is coherence-after the concurrent store of condition 1.
5. Complex, timing-sensitive micro-architectural conditions occur.

## Implications

If the conditions are met, the last load might forward data to dependent instructions that is inconsistent with the data written to the architectural register file. This load might forward the data from the store in condition 1 to dependent instructions, while writing the data of the store in step 4 to the architectural register file.

## Workaround

Software can set `IMP_CPUACTLR_EL1[4] = 1`; for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #4, #1
MSR S3_0_C15_C1_0, x0
```

This increases the best-case load-use penalty by one cycle. Some individual tests of the benchmarks can cause an IPC reduction of up to 9% (SPECint2006) and 6% (Geekbench v5), but the geometric mean reduction is of 3-4% (SPECint2006) and 2% (Geekbench v5).

## 2217821
## Heavy I-cache invalidation or TLB invalidation traffic from other PEs can cause a deadlock

### Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

### Description

Heavy I-cache invalidation or TLB invalidation traffic from other PEs in the system can cause a deadlock.

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under the following conditions:

1. One or more other PEs in the system generate sequences of Instruction Cache Invalidate or TLB Invalidate traffic without intervening `DSB` instructions.
2. Heavy memory traffic across the system occurs downstream of the L2 caches.
3. Complex microarchitectural timing conditions occur.

### Implications

If the conditions are met, the system might deadlock.

### Workaround

There is currently no workaround for this erratum.

# 2454944
# Unmodified cache line might be written back to memory

## Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

Cacheable store operations might incorrectly cause an unrelated cache line to be marked as modified, which later results in that line being written back to memory.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. Memory location A is marked as Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. The core allocates location A into the L1 data cache. This allocation might be due to committed instructions, Speculative execution or data prefetching.
3. The core executes a number of stores to a cacheable memory location other than location A.
4. Uncommon, timing-sensitive microarchitectural conditions occur.

## Implications

If the previous conditions are met, the cache line for memory location A might be marked as modified but the data remains unchanged. When the cache line is subsequently evicted, it will then overwrite the value in memory. If an agent in the system has made non-coherent writes to location A, these writes might then be overwritten with the older data from the core cache write-back, causing these writes to no longer be visible. This might then result in data corruption for software-managed coherency use cases.

Stores in condition 3 modify their cache lines as expected, this erratum cannot cause a modified cache line to still be marked as clean in the cache.

## Workaround

This erratum has no workaround.

# Category A (rare)

## 1965350
## Coherency might be lost on a cache line

### Status

Fault Type: Programmer Category A (rare)
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

Under rare circumstances, coherency may be lost on a cache line filled into the L1 data cache.

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under the following conditions:

1. A core performs an L1 cache refill.
2. The same core performs another L1 cache refill.
3. Rare, timing-sensitive micro-architectural conditions occur.

### Implications

If the conditions are met, coherency might be lost on the cache line filled into the L1 data cache.

### Workaround

There is no workaround.

# 1975996
# Non-L1-allocating stores might cause data corruption

## Status

Fault Type: Programmer Category A (rare)
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

In rare circumstances, stores might cause data corruption on the bytes written.

## Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache, and with two L2 RAM partitions per slice (configuration parameter L2_CACHE is TRUE and L2_NUM_RAMCTL_PARTITIONS is 2).

## Conditions

The erratum occurs under the following conditions:

1. Three or more non-L1-allocating stores (excluding stores to non-gathering Device memory) to the same 16-byte-aligned 16-byte granule are executed close to each other. Stores are non-L1-allocating if any of the following conditions is met:
    - The memory type is not Normal Inner Write-Back, Outer Write-Back Cacheable.
    - The memory attributes are Transient.
    - They are generated by `STNP` or `DC ZVA` instructions.
    - The core is in write-streaming mode.
2. Rare, timing-sensitive microarchitectural conditions occur.

## Implications

If the conditions are met, data in the 16-byte-aligned 16-byte granule that the stores are writing to might be corrupted.

## Workaround

There is no workaround.

# Category B

## 1902691
## Trace Buffer Extension can cause trace corruption or deadlock

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

### Description

The use of Trace Buffer Extension (TRBE) to write trace data from the ETM to memory can result in trace data corruption or deadlock.

### Configurations Affected

All configurations are affected.

### Conditions

1. TRBE is enabled by setting TRBLIMITR_EL1.E to 1.
2. The ETM is enabled and generating trace.

### Implications

TRBE should not be used on this revision of the product, despite being advertised as supported in the ID registers.

### Workaround

System firmware should disable the use of TRBE, by setting MDCR_EL3.NSTB to 0b00 and TRBLIMITR_EL1.E to 0.

## 1910738 Use of tagged memory might cause deadlock, data corruption or incorrect reporting

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If a memory location is mapped as tagged memory and MTE is enabled, then deadlock, data corruption, or incorrect reporting in TFSRE0_EL1 or TFSR_ELx might occur.

### Configurations Affected

This erratum affects configurations where the **BROADCASTMTE** pin is HIGH.

### Conditions

1. MTE is enabled (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn != 0b00).
2. A page is mapped as tagged memory.
3. Unlikely micro-architectural conditions involving the exact timing of multiple operations.

### Implications

If these conditions are met, then one or more of the following might occur:

- Deadlock
- Data corruption of either data or allocation tags
- Spurious fault reporting or failure to report faults in TFSRE0_EL1 or TFSR_ELx

### Workaround

This erratum can be avoided by setting IMP_CPUECTLR_EL1[19] = 1, IMP_CPUACTLR_EL1[4] = 1 and IMP_CPUACTLR_EL1[26] = 1, for example using the following code sequence:

```
MRS x0, S3_0_C15_C1_4
MOV x1, #1
BFI x0, x1, #19, #1
MSR S3_0_C15_C1_4, x0
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #4, #1
BFI x0, x1, #26, #1
MSR S3_0_C15_C1_0, x0
```

# 1922240
# SnpPreferUnique* might lead to a loss of coherency

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Coherency might be lost if the core receives a SnpPreferUnique or SnpPreferUniqueFwd as a result of either:

- a SnpPreferUnique or SnpPreferUniqueFwd external to the cluster, or
- a ReadPreferUnique or MakeReadUnique(excl) on another core in the cluster.

A ReadPreferUnique is typically triggered by load exclusive instructions as well as by speculative stores. A MakeReadUnique(excl) can only be triggered by a non-speculative store exclusive instruction.

## Configurations Affected

All configurations are affected.

## Conditions

1. A cache line A is present in the L1 data cache of at least one core in the complex.
2. The core is executing an exclusive sequence.
3. A SnpPreferUnique or SnpPreferUniqueFwd transaction to cache line A is received by the complex.

## Implications

If the conditions are met, coherency might be lost on cache line A.

## Workaround

This erratum can be avoided by setting IMP_CMPXACTLR_EL1[11:10] = 3; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_3
MOV x1, #3
BFI x0, x1, #10, #2
MSR S3_0_C15_C1_3, x0
```

This workaround has negligible performance impact.

# 1937669
# Tag Checked store might not be correctly ordered by DMB

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The Tag read for a Tag Checked store might not be ordered correctly by a **`DMB`**.

## Configurations Affected

This erratum affects configurations where the **BROADCASTMTE** pin is HIGH.

## Conditions

1. The core starts a memory access.
2. A **`DMB`** instruction that orders loads is executed (any **`DMB`** without the **`ST`** qualifier).
3. A Tag Checked store is executed.

## Implications

In the above sequence, the Tag read generated by the Tag Checked store instruction might not be ordered correctly with other memory accesses before the **`DMB`** instruction.

## Workaround

This erratum can be avoided by setting IMP_CPUACTLR_EL1[10] = 1; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #10, #1
MSR S3_0_C15_C1_0, x0
```

This upgrades **`DMB`** instructions to **`DSB`**, potentially incurring a small but not negligible performance cost.

# 1942494
# A cacheable load of SIMD&FP or SVE registers might return incorrect data

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A cacheable Neon, SVE, or FP load might return incorrect data under certain micro-architectural conditions.

## Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache (configuration parameter L2_CACHE is TRUE).

## Conditions

The following sequence is executed:

1. An unaligned load crossing a 16-byte boundary, an **LDG** instruction, or a Tag Checked load, to memory mapped as Normal Inner Write-Back, Outer Write-Back.
2. A load of one or more SIMD&FP or SVE registers that partially or fully overlaps the previous load, but does not cross a 16-byte boundary and is Tag Unchecked.

Both accesses miss in the L1 data cache, but hit in the L2 cache.

## Implications

If the above conditions are met, under some rare micro-architectural conditions, the second load might return incorrect data.

## Workaround

This erratum can be avoided by setting IMP_CPUACTLR_EL1[15] = 1; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #15, #1
MSR S3_0_C15_C1_0, x0
```

This increases the best case L2 hit latency by a cycle, incurring a small but not negligible performance cost.

# 1943339
# Cacheable far atomics might result in loss of coherency

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A cacheable atomic instruction might result in a loss of coherency if the cache line is present in the complex.

## Configurations Affected

This erratum affects all configurations.

## Conditions

- An atomic instruction to a cacheable location is executed on a core in the complex. The atomic instruction is executed as a "far" atomic (outside the core).
- The cache line is also present in the L1 data cache of at least one of the cores within the complex.
- A snoop is received by the complex for the cache line.

## Implications

If the conditions are met, coherency might be lost on the accessed cache line.

## Workaround

If a workaround is necessary, atomic instructions can be forced to execute "near" (in the core), rather than "far" by setting IMP_CPUECTLR_EL1[40:38] = 0b010; for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_4
MOV x1, #2
BFI x0, x1, #38, #3
MSR S3_0_C15_C1_4, x0
```

# 1952872
# A cacheable load might return incorrect data

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A cacheable load of general-purpose registers might return incorrect data under some microarchitectural conditions.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Unlikely microarchitectural conditions occur.
2. A cacheable load of one or more general-purpose registers is executed which hits in the L1 data cache.

## Implications

If the above conditions are met, incorrect data may be returned.

## Workaround

This erratum can be avoided by setting IMP_CPUACTLR_EL1[18] = 1, for example using the following code sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #18, #1
MSR S3_0_C15_C1_0, x0
```

This workaround may have a small but not negligible performance impact. The reach of the L1 data TLB is reduced in some cases.

# 1955641
# Core in FULL_RET might cause data corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

When one core in a complex is in the FULL_RET power mode and the other core is OFF, this might either cause data corruption or prevent reporting of RAS error interrupts.

## Configurations Affected

This erratum affects all configurations with two cores in a complex.

## Conditions

1. One core in a complex is in the OFF power mode.
2. The other core in the same complex enters the FULL_RET power mode.

## Implications

The complex will incorrectly remove itself from coherency with the rest of the cluster, leading to data corruption if another agent tries to snoop data that is in the complex.

If there is an existing error in the complex RAS registers, causing the **nCOMPLEXFAULTIRQ** and **nCOMPLEXERRIRQ** outputs to be asserted LOW, then these signals might be incorrectly deasserted HIGH while the complex is in FULL_RET. These errors will then not be visible to the system until the core leaves FULL_RET.

## Workaround

The FULL_RET power mode should not be enabled. This can be done by setting both IMP_CPUPWRCTLR_EL1.WFE_RET_CTL and IMP_CPUPWRCTLR_EL1.WFI_RET_CTL to 0b000, which is their default value.

# 1956116
# Missing trace for debug entry due to watchpoint hit

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If ETM tracing is enabled and a watchpoint hit occurs that is configured to enter debug state, the waypoint for the debug entry will not be traced.

## Configurations Affected

All configurations are affected.

## Conditions

The erratum occurs under the following conditions:

1. Halting debug is enabled (EDSCR.HDE == 1).
2. Halting is allowed.
3. The OS Lock is unlocked (OSLSR_EL1.OSLK == 0).
4. ETM tracing is enabled (TRCPRGCTLR.EN == 1). This can be enabled either by software or an external debugger.
5. Tracing in the current region is unprohibited.
6. An enabled watchpoint is hit.

## Implications

If these conditions are met, then no Exception element will be traced for the debug state entry.

## Workaround

There is no workaround for this erratum

## 1961781
## Watchpoints on SVE first-fault and non-fault loads might cause an incorrect value to be loaded and/or an incorrect MTE check results

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If a watchpoint is programmed on a byte accessed by a non-fault load, or by a first-fault load, and is on a byte not covered by the first element, a wrong value might be loaded on all active bytes. If MTE checking is enabled, the check results might be incorrect.

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under the following conditions:

1. A watchpoint is programmed on a byte A.
2. An SVE non-fault load is executed which loads from byte A on any element, or an SVE first-fault load is executed which loads from byte A on any non-first element.

### Implications

If these conditions are met, the data loaded in all lanes by the SVE load above might be incorrect.

If MTE checking is enabled and the instruction fails the MTE check, the FFR might be incorrect.

If MTE checking is enabled with SCTLR_ELx.TCF set to 0b01 (precise checking), an incorrect exception might be taken or no exception might be taken even if the tag check should have failed.

If MTE checking is enabled with SCTLR_ELx.TCF set to 0b10 (imprecise checking), an incorrect value might be reported in TFSR_ELx following the execution of the load.

This erratum has been categorized on the basis of use in sample systems only. A workaround is available where the use of watchpoints for memory locations accessed by SVE non-fault and first-fault instructions is required on sample silicon.

### Workaround

Software can avoid this erratum by programming the watchpoint to cover a 32-byte-aligned 32-byte quantity.

# 1962857
# Set/way maintenance of L2 cache might cause data corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Use of set/way maintenance instructions targeting the L2 cache might cause data corruption of the cache line targeted by the operation.

## Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache and contains two cores.

## Conditions

1. A cache line is present in both the L2 cache and one of the L1 data caches.
2. A `DC CISW`, `DC ISW`, or `DC CSW` instruction is executed, targeting the L2 cache set/way that contains the cache line above.
3. The complex receives a snoop, other than SnpOnce, targeting the same cache line. This can be caused by any memory access to that cache line, in the cluster or outside.
4. One of the cores in the complex starts an L1 data cache refill to the same L1 data cache set as the cache line above.

## Implications

If the conditions are met, data corruption may occur on the cache line targeted by the set/way maintenance instruction. The hardware flush mechanism is not affected.

## Workaround

Software should only use set/way maintenance instructions targeting the L2 cache when caching is disabled in both cores of the complex. Note that the core performs hardware cache invalidation on powerup and powerdown, so there is generally no need for explicit software cache maintenance by set/way.

To prevent problems with set/way maintenance instructions in virtual machines, hypervisors can trap set/way instructions using HCR_EL2.TSW=1.

## 1966377
## Incorrect instructions might be executed

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

Incorrect instructions might be executed.

### Configurations Affected

All configurations are affected.

### Conditions

The erratum occurs under the following conditions:

1. The core executes with MMU enabled.
2. Additional complex micro-architectural conditions occur.

### Implications

If the above conditions are met, the core might execute incorrect instructions. The effects of the incorrect execution are not predictable and include:

- Spurious UNDEFINED instruction exceptions.
- Spurious Pre-fetch Abort, Data Aborts, or other exceptions.
- Incorrect instruction execution, which might lead to register or memory state corruption.

### Workaround

This erratum can be avoided by setting IMP_CPUACTLR2_EL1[29] to 1 to disable a power optimization feature; for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1 << 29
MSR S3_0_C15_C1_1, x0
```

This will have no impact on performance, but will slightly increase power consumption (overall 1-2%).

# 1975068
# Cacheable Non-shareable mappings might lead to CHI protocol violations

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Hardware prefetches to memory mapped as Cacheable Non-shareable may result in CHI protocol violations, which in turn may result in deadlocks, hazarding failures and potentially loss of coherency in rare circumstances.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. A page of memory is mapped as Inner Write-Back, Outer Write-Back Non-shareable.
2. The interconnect returns a RespSepData response after DataSepResp has been sent for a hardware-prefetch-initiated ReadNoSnoop to the location above.

## Implications

If the conditions are met, the hardware prefetcher may cause a CHI protocol violation by reusing a TxnID early. This may result in deadlocks, hazarding failures and potentially loss of coherency in rare circumstances.

## Workaround

In some systems, software can avoid using Non-shareable mappings. Where that is not possible, software can set IMP_CMPXECTLR_EL1[9:8] = 0b11, for example using the following sequence:

```
MRS x0, S3_0_C15_C1_7
MOV x1, #3
BFI x0, x1, #8, #2
MSR S3_0_C15_C1_7, x0
```

This disables early forwarding of L2 hardware prefetches to subsequent requests, and may incur a small but not negligible performance impact.

## 1977575
## Coherency might be lost on a cache line shortly after a core powers up

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

When the MMU and caches are enabled almost immediately after a core is powered on, coherency might be lost on an arbitrary cache line.

### Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache, and with two cores.

### Conditions

1. One of the two cores in the complex is generating heavy cacheable L1 cache miss traffic.
2. The second core in the complex is powered on, and almost immediately enables the MMU and caches, and starts executing cacheable loads and stores.
3. Rare, timing-sensitive, microarchitectural conditions occur.

### Implications

If the conditions are met, coherency might be lost on an arbitrary cache line with PA[11:6] = 0x3f. The exact effects and cache line affected depends on the reset value of the duplicate tag RAMs.

### Workaround

Software can execute 4 loads to 4 different cache lines with a physical address (PA) that does not match PA[11:6] = 0x3f on the CPU that just powered up before doing any other access to Inner Write-back and Outer Write-back memory. These loads must be the first 4 demand load/stores to normal memory with Inner Write-back and Outer Write-back cacheability, just after turning on the MMU and data caches. The loads must be followed by a DSB SY.

# 1981843
# A store exclusive to shareable memory other than Normal Inner Write-Back, Outer Write-Back Cacheable might cause a hang

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A store exclusive to shareable memory other than Normal Inner Write-Back, Outer Write-Back Cacheable might hang.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core executes a store exclusive instruction to shareable memory other than Normal Inner Write-Back, Outer Write-Back Cacheable.
2. Unlikely, but not rare, timing-sensitive, microarchitectural conditions occur.

## Implications

If the conditions are met, the core executing the store exclusive instruction might hang. This erratum has been classified based on the understanding that the affected version of the product is only used for samples.

## Workaround

As the support for exclusives to memory other than Normal Inner Write-Back, Outer Write-Back Cacheable is implementation defined, programmers should behave as if this is not supported.

# 2008766
# RAS errors during core power down might cause a deadlock

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2 and r1p3. Open.

## Description

If a Reliability, Availability, and Serviceability (RAS) error occurs when a core is being powered down, then the power down sequence is designed to be aborted so that the error can be handled by software. As explained in this erratum, the power down sequence might complete despite the error occurring, or it might cause a deadlock.

## Configurations affected

All configurations are affected.

## Conditions

1. The ERXCTLR_EL1.ED bit is set for any of the core error records, and at least one of the CI, DUI, CFI, FI, or UI bits is set.
2. Software running on the core sets the CPUPWRCTLR.CORE_PWRDN_EN bit and executes a `WFI` instruction.
3. The core Power Policy Unit (PPU) requests that the core transitions from the ON power mode to either the OFF or OFF_EMU power mode.
4. During the L1 or L2 cache clean and invalidate done by the power transition, a RAS error occurs that causes any of the **nCOREFAULTIRQ**, **nCOREERRIRQ**, **nCOMPLEXFAULTIRQ**, or **nCOMPLEXERRIRQ** pins to be asserted.

## Implications

If the erratum occurs, the PPU power down request will either:

- Be denied because of the RAS error
- Deadlock, preventing the power down from completing.

If the power down request is denied, the core will not wake up from the `WFI` instruction and therefore software is not able to handle the error. If the PPU requests the core to power down again, either because it is set to dynamic mode, or because the component programming the PPU requests the power down again, then the second power down might either:

- Complete and power off the core, despite the fact that the error has not been handled.

- Deadlock, preventing the power down from completing.

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate because of this erratum.

## Workaround

The ERXCTLR_EL1.ED bit should be cleared for all the core error records before software sets the CPUPWRCTLR.CORE_PWRDN_EN bit to request a power down. This will cause any error detected during the power down to be ignored.

## 2014530
## Debug accesses while leaving OFF_EMU power mode might lead to a system deadlock

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

### Description

If a core in the complex leaves the OFF_EMU power mode while there is traffic on the debug APB interface to that core, or trace traffic on the ATB interface out of the core, then it can lead to a system deadlock or corruption of the trace data.

### Configurations Affected

This erratum affects all configurations with two cores in a complex where an asynchronous bridge is present between the complex and the cluster (parameter CORE<n>_ASYNC_BRIDGE is TRUE).

### Conditions

The erratum occurs under the following conditions:

1. One core in the complex is in the OFF_EMU power mode.
2. The core is requested to power ON.
3. While the power transition is in progress, an access is made on the Debug APB interface that targets a register in the core, or the core generates trace data on the ATB interface.

### Implications

If the APB bus access happens during a specific time window within the power transition, then it might not receive a response. This can lead to a system deadlock. If the core is sending trace data on the ATB interface during the same time window, then additional trace transactions might be sent which will corrupt the trace stream.

### Workaround

The debugger should not set the DBGPRCR.CORENPDRQ bit and the PPU should not be programmed to enter the OFF_EMU power mode.

# 2015106
# Memory mapped accesses to MPMM registers have the wrong offset

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The CPUMPMMCR register is documented as being located at offset 0x10 within the MPMM register block. Due to this erratum, the actual location is 0x90.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. A memory mapped access is made to the CPUMPMMCR register.

## Implications

If the conditions are met, the CPUMPMMCR will not actually be accessed.

## Workaround

Software accessing the CPUMPMMCR on revisions affected by this erratum should use the offset 0x90.

# 2022138
# Core in FULL_RET might not have clock gated

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

When a core enters the FULL_RET power mode due to a Wait for Event (WFE) or Wait for Interrupt (WFI) instruction, and at the same time an event or interrupt arrives to wake up the WFE or WFI, this might prevent the clock from being gated correctly when in the FULL_RET power mode.

## Configurations Affected

This erratum affects all configurations if the FULL_RET power mode has been implemented for logic retention.

## Conditions

1. The IMP_CPUPWRCTLR_EL1.WFE_RET_CTL or IMP_CPUPWRCTLR_EL1.WFI_RET_CTL register is set to a non-zero value.
2. A core is in either the FUNC_RET or ON power modes.
3. A WFE or WFI instruction is executed.
4. After the time period configured in the IMP_CPUPWRCTLR_EL1.WFE_RET_CTL or IMP_CPUPWRCTLR_EL1.WFI_RET_CTL register, the core will transition to the FULL_RET power mode.
5. During the transition, an event arrives from another core or component in the system which causes the core to leave WFE or WFI.

## Implications

The core might enter the FULL_RET power mode for a short period before the event causes a transition out of FULL_RET. However, during the time the core is in FULL_RET the clock to the core will toggle when the logic is in a retention state. This might cause undesirable physical effects in the power domain, including damaging the device.

## Workaround

The FULL_RET power mode should not be enabled for WFE or WFI. This can be done by setting IMP_CPUPWRCTLR_EL1.WFE_RET_CTL to 0b000 and IMP_CPUPWRCTLR_EL1.WFI_RET_CTL to 0b000, which are the default values. EL3 software should not set ACTLR_EL3.PWREN so that lower exception levels cannot enable this functionality.

# 2027318
# SnpPreferUnique might lead to a loss of coherency

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

Coherency might be lost if the core receives a SnpPreferUnique as a result of either:

- a SnpPreferUnique or SnpPreferUniqueFwd external to the cluster, or
- a ReadPreferUnique or MakeReadUnique(excl) on another core in the cluster.

A ReadPreferUnique is typically triggered by load exclusive instructions and by speculative stores. A MakeReadUnique(excl) can only be triggered by a non-speculative store exclusive instruction.

## Configurations Affected

This erratum affects configurations with two cores in a complex.

## Conditions

1. A cache line A is present in the L1 data cache of both cores in the complex, but is not in a unique state.
2. One core is executing an exclusive sequence.
3. Another core is evicting cache line A from its L1 data cache.
4. The complex receives a SnpPreferUnique transaction to cache line A.

## Implications

If the conditions are met, coherency might be lost on cache line A.

## Workaround

This erratum can be avoided by setting IMP_CMPXACTLR_EL1[11] = 0b1; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_3
MOV x1, #1
BFI x0, x1, #11, #1
MSR S3_0_C15_C1_3, x0
```

This workaround has negligible performance impact.

# 2028010
# A PRF{U}M instruction might hang

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

A **PRF{U}M** instruction might hang if hardware updates of the dirty bit are enabled.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. Three or more **PRF{U}M PLDL1*/PSTL1*** instructions are executed. At least two miss in the L1 data Translation Lookaside Buffer (D-TLB).
2. Two stores or atomics are executed to different 4KB regions of virtual memory, and hardware updates of the dirty bit are enabled for both regions.
3. Another **PRF{U}M PLDL1*/PSTL1*** instruction is executed.

## Implications

If the conditions are met, a **PRF{U}M** instruction might hang indefinitely. Interrupts will not be taken.

## Workaround

Software can set IMP_CPUACTLR_EL1[38] = 1; for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #38, #1
MSR S3_0_C15_C1_0, x0
```

This is not expected to have a material performance impact in common use cases, but might lead to some **PRF{U}M PLDL1*/PSTL1*** instructions not prefetching the cache line into the L1 data cache.

# 2038114
# EDSCR.STATUS might be incorrect when single stepping a Load-Exclusive instruction

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

When in the Active-not-pending state, if a Load-Exclusive instruction generates an abort that takes an exception to a state where halting is allowed, then EDSCR.STATUS will be incorrect when entering Debug State.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. Halting step is enabled and is in the Active-not-pending state (EDECR.SS == 1 and EDESR.SS == 0).
2. A Load-Exclusive instruction is executed. Execution of this instruction causes an exception that targets a state where halting is allowed.
3. The core enters debug state.

## Implications

If the above conditions are met, then EDSCR.STATUS will be set to "Halting Step, Normal" upon debug entry.

## Workaround

There is no workaround.

# 2038923
# A change to TRBLIMITR_EL1.E might result in trace buffer corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

A change to TRBLIMITR_EL1.E might cause an inconsistent view on whether trace is prohibited within the CPU. As a result, the trace buffer or trace buffer state might be corrupted.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. The Trace Buffer Extension (TRBE) is enabled by setting TRBLIMITR_EL1.E = 0b1.
2. Only a single context synchronization event takes place before execution changes from a context in which trace is prohibited to one where it isn't, or vice versa.

## Implications

If the above conditions are met, the view of whether trace is prohibited is inconsistent between parts of the CPU, and the trace buffer or the trace buffer state might be corrupted.

## Workaround

Software can prevent an inconsistent view of whether trace is prohibited or not based on TRBLIMITR_EL1.E by immediately following a change to TRBLIMITR_EL1.E with at least one **`ISB`** instruction before an **`ERET`**, or two **`ISB`** instructions if no **`ERET`** is to take place.

## 2039165
## Debug accesses while leaving OFF_EMU power mode might lead to a system deadlock

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

### Description

If a core in the complex leaves the OFF_EMU power mode while there is traffic on the debug APB interface to that core, or trace traffic on the ATB interface out of the core, then it can lead to a system deadlock or corruption of the trace data.

### Configurations Affected

This erratum affects all configurations with two cores in a complex where an asynchronous bridge is not present between the complex and the cluster (parameter CORE<n>_ASYNC_BRIDGE is FALSE).

### Conditions

The erratum occurs under the following conditions:

1. One core in the complex is in the OFF_EMU power mode.
2. The core is requested to power ON.
3. While the power transition is in progress, an access is made on the Debug APB interface that targets a register in the core, or the core generates trace data on the ATB interface.

### Implications

If the APB bus access happens during a specific time window within the power transition, then it might not receive a response. This can lead to a system deadlock. If the core is sending trace data on the ATB interface during the same time window, then additional trace transactions might be sent which will corrupt the trace stream.

### Workaround

The debugger should not set the DBGPRCR.CORENPDRQ bit and the PPU should not be programmed to enter the OFF_EMU power mode.

# 2041661
# Entry into the Full Retention power mode might cause incorrect execution of a TLB maintenance instruction

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

If all cores in the complex enter the Full Retention power mode (or one core is Off and the other enters Full Retention mode), and the complex shared logic enters the FULL_RET power state, an outstanding TLB maintenance instruction (**`TLBI`**) from another PE in the system might not invalidate all affected L2 TLB entries in the complex.

## Configurations Affected

This erratum affects all configurations. It requires the implementer to have implemented FULL_RET support for the complex shared logic, and to place the L2 TLB RAMs into a low-power mode when in FULL_RET.

## Conditions

1. All cores in the complex are in the Full Retention power mode.
2. The complex receives a TLB maintenance operation from another PE outside the complex just before entering FULL_RET for the shared logic, or during FULL_RET of the shared logic.
3. The L2 TLB RAMs are placed into a low-power mode during the FULL_RET complex power state.

## Implications

If the conditions are met, the **`TLBI`** operation might not correctly invalidate all affected TLB entries. This erratum has been classified based on the understanding that the affected versions of the product are only used for samples.

## Workaround

This erratum can be avoided by disabling use of the Full Retention power mode in the cores (setting IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL to 0b000 and IMP_CPUPWRCTRL_EL1.WFE_RET_CTRL to 0b000).

Implementers may be able to avoid this erratum by not placing the L2 TLB RAMs into a low-power mode when the complex logic is in the FULL_RET power state, such that the outputs of these RAMs will remain stable while the clock is gated.

# 2041909
# Load-Exclusive/Store-Exclusive loop might not make forward progress

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

A Load-Exclusive/Store-Exclusive sequence might not make forward progress if loads or stores to addresses in the same L1 data cache set are present before the Load-Exclusive, and other PEs in the system execute stores to those cache lines.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The core executes a Load-Exclusive/Store-Exclusive loop to an address A. Address A is Normal Inner Write-back, Outer Write-back Cacheable.
2. Loads or stores to the same L1 data cache set as address A are executed between a failing Store-Exclusive and the next retry of the Load-Exclusive. Loads or stores are to the same set as an address A, if the virtual address bits [12:6] are the same.
3. One or more other PEs in the same shareability domain continuously execute stores to the same cache lines as the loads or stores in point 2.

## Implications

If the conditions are met, the exclusive sequence might fail to make forward progress until other PEs in the system stop storing to the same cache line as the loads or stores preceding the retry of the Load-Exclusive. For user code sequences, it can be assumed that stores by a different agent will stop in finite time, possibly as a result of an interrupt. For privileged code, this could be blocking to execution indefinitely.

## Workaround

Privileged software can work around this erratum by avoiding any loads and stores between a Store-Exclusive returning a failing result and the retry of the corresponding Load-Exclusive.

From the r0p2 version, it is possible to prevent this erratum from occurring by executing the following instruction sequence:

```
MOV     x0, #0
MSR     S3_6_C15_C4_0, x0
ISB

MOV     x0, #0x08500000
MSR     S3_6_C15_C4_2, x0

MOV     x0, #0x1F700000
MOVK    x0, #0x8, LSL #32
MSR     S3_6_C15_C4_3, x0

MOV     x0, #0x03F1
MOVK    x0, #0x0110, LSL #16
MSR     S3_6_C15_C4_1, x0

ISB
```

This will apply additional memory ordering to Load-Exclusive, Load-Acquire, and Load-AcquirePC instructions, and will cause a small performance loss.

If the software workaround is used, then the workaround should be disabled before hardware single stepping by executing the following instruction sequence:

```
MOV x0, #0
MSR S3_6_C15_C4_1, x0
ISB
```

The workaround should be re-applied before exiting debug state by executing the following instruction sequence:

```
MOV x0, #0x03F1
MOVK x0, #0x0110, LSL #16
MSR S3_6_C15_C4_1, x0
ISB
```

Note that the disabling and re-application of the workaround are only applicable if the debugger has access to EL3.

# 2042739
# A store or Load-Exclusive might cause data corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

A Store or Load-Exclusive instruction might cause data corruption on the cache line accessed.

## Configurations Affected

This erratum affects configurations with two cores in the complex, and where the complex is configured with an L2 cache and with two L2 RAM partitions per slice (configuration parameter L2_CACHE is TRUE and L2_NUM_RAMCTL_PARTITIONS is 2).

## Conditions

This erratum occurs under the following conditions:

1. The MMU is enabled.
2. A core executes a Store or Load-Exclusive to Normal Inner Write-back, Outer Write-back Cacheable memory for a cache line A. This might cause a ReadPreferUnique to be generated.
3. Cache line A is already present in the L1 data cache of the core, but under a different virtual address alias.
4. In addition to the L1 cache of the requester, the cache line is also present in the L2 cache. This can only occur if the other core in the complex also had the cache line, and has evicted it to the L2 cache.
5. Unlikely, timing-sensitive micro-architectural conditions occur.

## Implications

If the conditions are met, the data and MTE allocation tags on cache line A might become corrupted.

## Workaround

Software can set IMP_CPUECTLR_EL1[19] = 1 before the MMU is enabled; for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_4
MOV x1, #1
BFI x0, x1, #19, #1
```

```
MSR S3_0_C15_C1_4, x0
```

This workaround disables generation of ReadPreferUnique. It is expected to have a negligible performance impact.

# 2044951
# Entry into FUNC_RET power mode can cause data metastability

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

When the complex transitions from ON to FUNC_RET, some of the signals into the associated cores may toggle asynchronously and thus cause metastability in the associated cores' architectural state.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. Entry into the FUNC_RET power mode is enabled (IMP_CPUPWRCTLR_EL1.VPU_PWR_CTRL != 0b000) in all cores in the complex).
2. An FP, Advanced SIMD, or SVE instruction writing to a general-purpose destination register or PSTATE.NZCV is executed speculatively and flushed.
3. No FP, Advanced SIMD, or SVE instructions are issued for the period of time configured in IMP_CPUPWRCTLR_EL1.VPU_PWR_CTRL, such that the complex moves into the FUNC_RET power state.

## Implications

If these conditions are met, then some of the signals from the shared Vector Processing Unit (VPU) logic in the complex that are sampled in the associated cores may change asynchronously. This could cause metastability in core logic that controls updates to scalar general-purpose registers and PSTATE.NZCV. Theoretically, this could result in this state becoming corrupted.

## Workaround

The FUNC_RET power mode should not be used. This requires that IMP_CPUPWRCTLR_EL1.VPU_PWR_CTRL should be 0b000. This is the default value for this register.

## 2051678
## A hardware update of the dirty bit might not be correctly ordered with respect to later instructions

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

### Description

A hardware update of the dirty bit might not be correctly ordered with respect to later instructions, where ordering is required by the architecture.

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under the following conditions:

1. A load is executed to Virtual Address (VA) A.
2. A store or atomic is executed to a VA B.
3. VA B is not to the same 4KB region as VA A, but one of the following conditions is met:
    - 4KB descriptors are used for both VA A and VA B, and both VAs are within the same 16KB region. The descriptor for VA B is concurrently modified without break-before-make to have a different set of access permission bits.
    - Only a single page/block descriptor with a size larger than 4KB is used to map both VA A and VA B. The descriptor is concurrently modified without break-before-make to have a different set of access permission bits.
    - VA B is mapped via two or more descriptors of different sizes with a different set of access permission bits, with at least one of the mappings not including write permissions.
4. The core executes either:
    - A `DSB`, followed by a load, store, or cache maintenance operation
    - A store or atomic to the translation table entry for VA B

### Implications

If the conditions are met, the access permissions of the descriptor for VA B might be updated to include write permissions (dirtied), but this update might not be ordered with respect to the subsequent memory access after the `DSB`, or the subsequent store or atomic to the translation table entry that is being updated.

## Workaround

Software can disable the hardware update of the dirty bit by setting TCR_ELx.HD to 0b0 and VTCR_EL2.HD to 0b0.

# 2061513
# TRBE might cause a speculative hardware update of the dirty bit outside the trace buffer range

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

If the Embedded Trace Extension (ETE) is enabled, and the Trace Buffer Extension (TRBE) is enabled, a speculative hardware update of the dirty bit might occur to the first three pages of the virtual address space.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. TRBE is enabled by setting TRBLIMITR_EL1.E to 1.
2. TRBLIMITR_EL1.LIMIT is set to 0xffff_ffff_ffff_d, 0xffff_ffff_ffff_e, or 0xffff_ffff_ffff_f (the limit pointer address is set within 24KB of the top of the address space).
3. Hardware updates of the dirty bit are enabled for at least one of the first three pages of the address space (virtual address range 0x0000_0000_0000_0000-0x0000_0000_0000_2fff).

## Implications

If the conditions are met, a speculative hardware update of the dirty bit might occur to one of the pages in the virtual address range 0x0000_0000_0000_0000-0x0000_0000_0000_2fff. For most systems, having the dirty bit speculatively updated will not lead to any incorrect operation.

## Workaround

The software might be able to avoid setting the upper limit of the trace buffer within 24KB of the top of the address space.
Another workaround consists in disabling hardware update of the dirty bit for the first three pages of the address space.

# 2064142
# A system register write might not take effect after the trace buffer is disabled

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

If the Embedded Trace Extension (ETE) is enabled, and the Trace Buffer Extension (TRBE) is enabled after it is disabled, a register write might be ignored.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. TRBE is enabled and running.
2. The trace buffer is disabled, by setting TRBLIMITR_EL1.E = 0b0.
3. The core executes an **`MSR`** instruction to one of the following registers without first executing a **`TSB CSYNC`** and **`DSB`**:
    - TRBLIMITR_EL1
    - TRBPTR_EL1
    - TRBBASER_EL1
    - TRBSR_EL1
    - TRBTRG_EL1

## Implications

If the conditions are met, the register write in the last condition might be ignored and not take effect.

## Workaround

Software can execute a **`TSB CSYNC`** and **`DSB`** after collection is stopped and before performing a system register write to one of the affected registers.

## 2069954
## A hardware update of the dirty bit might occur speculatively if PSTATE.PAN is used

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

### Description

A hardware update of the dirty bit might be speculatively performed for a store or atomic instruction that sees a permission fault because of Privileged Access Never (PAN).

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under two sets of conditions:

Set 1:

1. The core is executing in EL2 with Virtualization Host Extensions (VHE) enabled (HCR_EL2.E2H = 0b1).
2. PSTATE.PAN is set.
3. A store or atomic instruction is executed which sees a permission fault because of PAN.
4. Hardware update of the dirty bit is enabled for the page of memory accessed by the store or atomic instruction above.

Set 2:

1. The core is executing in EL1.
2. PSTATE.PAN is set.
3. A store or atomic instruction is executed which sees a permission fault because of PAN.
4. The descriptor's AP[1] bit was concurrently modified from no EL0 access to allowing EL0 access.
5. Hardware update of the dirty bit is enabled for the page of memory accessed by the store or atomic instruction above.

### Implications

If the conditions are met, the access permissions of the descriptor might be speculatively updated to include write permissions (dirtied), even though the store sees a permission fault because of PAN. For most systems, having the dirty bit speculatively updated in the presence of a permission failure does not lead to any incorrect operation.

## Workaround

Software can disable the hardware update of the dirty bit by setting TCR_ELx.HD = 0b0.

## 2077057
## Software stepping ERETAA or ERETAB might corrupt SPSR_ELx

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

### Description

A pointer authentication failure or trap when software stepping `ERETAA` or `ERETAB` in the active-not-pending state while PSTATE.BTYPE is non-zero might corrupt the value of SPSR_ELx upon taking the exception for the pointer authentication failure.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs under the following conditions:

1. SPSR_ELy.BTYPE is non-zero.
2. An exception return targeting ELz is executed at ELy, which enables software stepping (entering the active-not-pending state).
3. `ERETAA` or `ERETAB` is executed outside of a guarded page, and one of the following occurs, taking an exception to ELx:
    - The associated PAC operation fails.
    - The instruction is trapped by SCR_EL3.API or HCR_EL2.API

ELx, ELy, and ELz can be distinct or overlapping.

### Implications

If the conditions are met, then the value of SPSR_ELx after taking the PAC exception will be corrupted.

The following SPSR_ELx fields can be affected: TCO, DIT, UAO, PAN, IL, SSBS, D, A, I, F, and M [3:0].

The value written to the affected SPSR_ELx fields after taking the exception in condition 3 will take the value of the respective SPSR_ELz fields prior to condition 3. This implies that if EL2 is stepping EL1, and the pointer authentication exception is taken to EL2, this allows EL1 to provide an arbitrary SPSR_EL2 value.

### Workaround

To prevent EL3 taking an exception with a corrupted SPSR_EL3:

- EL3 must set SCR_EL3.API to disable traps from lower exception levels.

When EL2 is single-stepping EL1 and HCR_EL2.API is configured to trap instructions:

- EL2 must keep the SPSR_EL2 value when returning to the active-not-pending state. If the next exception is a trap due to HCR_EL2.API, the SPSR_EL2 value may be corrupted, and the previously saved value should be used instead.

# 2082334
# ERR2STATUS.SERR might be incorrect

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

The value of ERR2STATUS.SERR uses 0x15 instead of 0x12, and vice versa. Other ERR2STATUS fields are not affected.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. This erratum occurs whenever the complex sees a non-attributable NDErr or DErr response.

## Implications

If the condition is met, then the ERR2STATUS register is updated correctly, except for the ERR2STATUS.SERR field. The ERR2STATUS.UE bit will indicate that an uncorrected error occurred.

## Workaround

Software can choose to translate the values 0x15 to 0x12 and vice versa when reading ERR2STATUS.SERR on affected hardware.

# 2135898
# DSPSR_EL0.BTYPE might be incorrect on Debug state entry

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

Entry to debug state might set DSPSR_EL0.BTYPE to the incorrect value.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. An indirect branch instruction is executed and sets PSTATE.BTYPE non-zero. The branch will be one of the following instructions:
   - BR
   - BRAA
   - BRAAZ
   - BRAB
   - BRABZ
   - BLR
   - BLRAA
   - BLRAAZ
   - BLRAB
   - **BLRABZ**
2. Another instruction is executed that sets PSTATE.BTYPE to zero. The instruction does not cause a synchronous exception.
3. Complex, but not rare, microarchitectural conditions occur.
4. The core enters Debug state.

## Implications

If these conditions are met, then the value of DSPSR_EL0.BTYPE will take the value of PSTATE.BTYPE set by the branch in step 1.

## Workaround

After entering debug state a debugger should set DSPSR_EL0.BTYPE to zero.

# 2141267
# LD1R using SP as base register incorrectly requests MTE Tag Checked

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

As specified in the Arm Architecture Reference Manual section D6.8.1, instructions using an immediate offset addressing form with SP as the base register must treat the instruction as Tag Unchecked. This erratum causes SVE LD1R load and broadcast to vector instructions to be treated as Tag Checked when used with SP as the base register.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs when all the following conditions occur:

1. The system is configured with support for the FEAT_MTEv2 feature.
2. Access to allocation tags at the current Exception level is enabled using SCR_EL3.ATA, HCR_EL2.ATA, SCTLR_ELx.ATA, and SCTLR_ELx.ATA0 as appropriate.
3. Access to SVE instructions at the current Exception level is enabled using CPTR_EL3.EZ, CPTR_EL2.ZEN, and CPACR_EL1.ZEN as appropriate.
4. PSTATE.TCO is 0.
5. An LD1RB, LD1RSB, LD1RH, LD1RSH, LD1RW, LD1RSW, or LD1RD broadcast to vector instruction is executed.
6. The base register used in the instruction is SP.
7. A virtual address accessed by the instruction has the Stage 1 translation Tagged memory attribute and it is not removed by a Stage 2 translation.
8. Bits [59:56] of the value in SP are not 0, or the TCR_ELx.TCMAy bit for the virtual address accessed is 0.
9. Bits [59:56] of the value in SP do not match the Allocation Tag associated with the memory locations accessed.
10. Handling of Tag Checked faults for the access, configured by SCTLR_ELx.TCF or SCTLR_ELx.TCF0 as appropriate, does not cause the fault to be ignored.

## Implications

If the erratum occurs, the processor will incorrectly generate a synchronous exception or set a bit in a TFSR_ELx or TFSRE0_ELx register. In a typical usage model, this will cause the system software to consider the executing software as behaving badly and might cause it to terminate execution of that software.

## Workaround

This erratum will only occur in software that has been built to:

- use the affected SVE instructions
- use FEAT_MTEv2 feature for checking stack accesses
- rely on the SP+imm instruction forms not being Tag Checked instructions

At the time of publishing, there are no tools which generate software where all three of the above conditions are true. It is recommended that tools building software which might be run on the affected IP do not cause all three conditions to be true.

# 2148497
# A trace unit trigger might result in inconsistent register state of the Trace Buffer Extension

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

If both the Embedded Trace Extension and the Trace Buffer Extension (TRBE) are enabled, a PE trace unit trigger might not update TRBSR_EL1.S.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. TRBE is enabled and running.
2. The TRBE is programmed to generate an IRQ on trigger (`TRBLIMITR_EL1.TM = 0b01`) or to stop on trigger (`TRBLIMITR_EL1.TM = 0b00`).
3. The core is executing in a prohibited trace region, and executes a `TSB CSYNC`.
4. The PE trace unit generates a trigger while the `TSB` instruction has not completed yet.
5. The `TSB CSYNC` completes, and the core turns off the TRBE (`TRBLIMITR_EL1.E = 0b0`) afterwards.
6. The core executes an `ISB` instruction, followed by a read of TRBSR_EL1 within 20 core cycles of the `TSB` completing.

## Implications

If the conditions are met, the TRBSR_EL1.S bit might incorrectly read as 0b0.

## Workaround

Software might be able to execute two `ISB` instructions instead of one `ISB` instruction between the disable of the TRBE and reading TRBSR_EL1.

## 2156527 Mapping memory as Tagged when Allocation Tag storage is not provided might lead to CHI protocol violations or Tag corruption

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

### Description

If a page is mapped as Tagged, but the underlying tag physical address (PA) does not provide Allocation Tag storage, stores to that page might lead to violations of the CHI protocol or Tag corruption.

### Configurations Affected

This erratum affects all configurations where the **BROADCASTMTE** pin is HIGH.

### Conditions

A page is mapped as Tagged, but Allocation Tag storage for that tag PA is not provided, and one of the following set of conditions is met:

Set 1:

1. MTE checking is enabled, and set to generate asynchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b10).
2. The core executes a checked store, and the store does not allocate to the L1 cache, for example because of hardware write streaming detection.

Set 2:

1. The core executes a store to tag memory, such as **`STG`**, **`STGM`**, **`STZG`**, **`ST2G`**, **`STZ2G`**, **`STGP`**, but not **`DC GZVA`**.
2. The store skips L1 allocation, for example because of hardware write streaming detection.
3. A snoop is received by the complex for the same PA while the store is outstanding.

### Implications

If the first set of conditions is met, a CHI protocol violation might occur.
If the second set of conditions is met, Tag corruption might occur.

### Workaround

Software should not map pages which do not provide Allocation Tag storage as Tagged.

# 2169012
# SnpPreferUnique* might lead to a loss of coherency

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

Coherency might be lost if a CPU core receives an SnpPreferUnique or SnpPreferUniqueFwd as a result of either:

- an SnpPreferUnique or SnpPreferUniqueFwd external to the cluster, or
- a ReadPreferUnique or MakeReadUnique(excl) on another core in the cluster.

A ReadPreferUnique is typically triggered by load exclusive instructions as well as by speculative stores. A MakeReadUnique(excl) can only be triggered by a non-speculative store exclusive instruction.

## Configurations Affected

All configurations are affected.

## Conditions

The erratum occurs under the following sets of conditions.

Set 1:

1. A cache line A is present in the L1 data cache of both cores in a CPU complex.
2. A core in the complex executes an L1 set/way operation for cache line A, or the core initiates a hardware-based L1 set/way operation due to a detected ECC error.
3. The complex receives an SnpPreferUnique or SnpPreferUniqueFwd snoop to cache line A while one of the CPUs in the complex is executing an exclusive sequence on that cache line.
4. The complex receives a second SnpPreferUnique or SnpPreferUniqueFwd snoop to cache line A before the set/way maintenance operation has completed.

Set 2:

1. A cache line A is present in the L1 data cache of both cores in a CPU complex.
2. A core in the complex executes a store exclusive, or an atomic instruction.
3. The complex receives an SnpPreferUnique or SnpPreferUniqueFwd snoop to cache line A while one of the CPUs in the complex is executing an exclusive sequence on that cache line.

## Implications

If the conditions are met, coherency might be lost on cache line A.

## Workaround

This erratum can be avoided by setting IMP_CMPXACTLR_EL1[11:10] = 0b11; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_3
MOV x1, #3
BFI x0, x1, #10, #2
MSR S3_0_C15_C1_3, x0
```

This workaround has negligible performance impact.

# 2172148
# Snp*UniqueFwd might cause a deadlock or data corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

An SnpUniqueFwd or SnpPreferUniqueFwd snoop received by the complex might cause a deadlock. These snoop operations are routinely generated as part of normal cacheable traffic.

## Configurations Affected

This erratum affects configurations with two cores in the complex.

## Conditions

1. MTE allocation tags of a cache line A have been written by a core in the same shareability domain as the complex.
2. The cache line A is present in both cores in the complex with dirty MTE tags.
3. The cache line is evicted from one of the cores in the complex.
4. The complex receives an SnpOnce or SnpOnceFwd to cache line A.
5. The cache line is evicted from the other core in the complex, and the hardware predicts that the cache line is unlikely to be reused in the near future. The complex generates a WriteClean request as a result.
6. The complex receives an SnpUniqueFwd or SnpPreferUniqueFwd for cache line A while the WriteClean is still outstanding.

## Implications

If the conditions are met, the complex might not send a CompData response, and instead send a spurious SnpRespData to the HN in response to the SnpUniqueFwd or SnpPreferUniqueFwd. This is a protocol violation, which might result in deadlock or data corruption of an arbitrary cache line.

## Workaround

This erratum can be avoided by forcing L2 allocation of transient lines, setting IMP_CPUECTLR_EL1[24:23] = 0b01 and IMP_CPUECTLR_EL1[47:46] = 0b01; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_4
MOV x1, #1
```

```
BFI x0, x1, #23, #2
BFI x0, x1, #46, #2
MSR S3_0_C15_C1_4, x0
```

This workaround has negligible performance impact.

## 2218134
## Single-bit error in branch predictor memory can cause instruction fetch stream corruption

### Status

Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

### Description

If a single-bit error occurs in the branch predictor memory, the instruction fetch stream might get corrupted.

### Configurations Affected

This erratum affects configurations with AARCH32_AT_EL0 parameter set to TRUE.

### Conditions

This erratum occurs under the following conditions:

1. The core executes with MMU enabled.
2. Core is in AArch32 execution state.
3. The lower 32 bits of the target address of a branch are the same in both AArch64 and AArch32 execution states, while the upper bits (48:32) of the virtual address are different.
4. A single-bit error occurs in the branch predictor memory.

### Implications

Instruction fetch stream might be irrecoverably corrupted.

### Workaround

Setting bit 43 of the IMP_CPUACTLR2_EL1 register (CPU Auxiliary Control Register 2) to 1 allows the instruction fetch stream to be corrected, with no performance impact.
Example code sequence:

```
MRS x0, S3_0_C15_C1_1
MOV x1, #1
BFI x0, x1, #43, #1
MSR S3_0_C15_C1_1, x0
```

## 2218950
## Aliased loads and stores might cause an ordering violation

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

### Description

A store followed by aliased loads might result in an ordering violation on one of the loads.

### Configurations Affected

This erratum affects all configurations.

### Conditions

The erratum occurs under the following conditions:

1. The core executes a store to a memory location A via a virtual address (VA) X.
2. The core executes two loads within three instructions of each other, both to memory location A. The younger load is to VA X, while the older load is to a different VA Y.
3. VA X and VA Y only differ in bits 15 and/or 14 of the address, and have identical memory attributes and permissions.
4. Another PE in the system concurrently modifies memory location A.
5. Another load, store, or atomic operation follows the younger load above within 8 instructions, and one of its address operands uses the register value loaded by the younger load above.

### Implications

If the conditions are met, the younger load at condition 2 may temporarily observe older data than the older load will retire with. This temporarily observed data can make it onto the forwarding paths within the core, and cause corruption of the address operands of the load, store, or atomic operation in condition 5. The value written into the register file is unaffected and will reflect the correct memory ordering.

### Workaround

Software can set `IMP_CPUACTLR_EL1[18] = 0b1` and `IMP_CMPXACTLR_EL1[25] = 0b1` for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
```

```
BFI x0, x1, #18, #1
MSR S3_0_C15_C1_0, x0

MRS x0, S3_0_C15_C1_3
MOV x1, #1
BFI x0, x1, #25, #1
MSR S3_0_C15_C1_3, x0
```

The workaround might have an impact on the reach of the L1 TLB and L2 TLB, impacting performance by approximately 1%.

# 2250311 Asynchronous exceptions while MPMM is active might corrupt processor state

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

The Maximum Power Mitigation Mechanism (MPMM) is a power management feature that detects and limits high activity events. If the count of high-activity events exceeds a pre-defined threshold during an evaluation period, MPMM temporarily limits the execution of instructions. Under certain micro-architectural conditions, asynchronous exceptions that arrive while MPMM is throttling the execution of instructions might be taken incorrectly.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. MPMM is enabled and is throttling instruction execution.
2. An asynchronous exception occurs.
3. Certain specific micro-architectural conditions occur.

r1p0 has the following additional condition:

1. The Core executes one of the following instruction sequence:
   - `AESD` followed by `AESIMC`.
   - `AESE` followed by `AESMC`.

## Implications

The exception will be taken incorrectly and might have the following effects:

- Exception level might be incorrect (The exception might be taken to EL0).
- PC might be incorrect.
- CPSR.M might be incorrect.
- ESR_EL1, ESR_EL2 or ESR_EL2 might be incorrect.
- FAR_EL1, FAR_EL2 or FAR_EL2 might be incorrect.

## Workaround

This erratum can be avoided by clearing bit 0 of IMP_CPUMPMMCR_EL3 to disable MPMM by using the following code sequence:

```
MRS x0, S3_6_C15_C2_1
BFC x0, #0, #1
MSR S3_6_C15_C2_1, x0
```

# 2256538
# Hardware breakpoints might not be taken on 32-bit T32 instructions

## Status

Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

## Description

Under certain micro-architectural conditions, hardware breakpoints might not be indicated on 32-bit T32 instructions.

## Configurations Affected

This erratum affects configurations with AARCH32_AT_EL0 parameter set to TRUE.

## Conditions

This erratum occurs under the following conditions:

1. Any of the following conditions are programmed for the hardware breakpoint:
    - The breakpoint is set on a 32-bit word-aligned T32 instruction, with the Byte Address Strobe (BAS) field set to 0b1111.
    - The breakpoint is set on a 32-bit halfword-aligned T32 instruction which is not word-aligned.
2. Micro-architectural conditions occur.

## Implications

For 32-bit T32 instructions, hardware breakpoints might not be indicated.

## Workaround

For word-aligned 32-bit T32 instructions, the Byte Address Strobe (BAS field of the DBGBCR register) should be set to 0b0011, which is the recommended value for this scenario.

For halfword-aligned 32-bit T32 instructions, there is no reliable workaround. A debugger might be able to make use of the software breakpoint instruction or program the hardware breakpoint to a nearby (word-aligned) instruction.

## 2288014
## Data corruption might occur in rare circumstances

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

### Description

Data corruption might occur in rare circumstances.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. Rare, timing-sensitive, microarchitectural conditions occur.

### Implications

If the conditions are met, data corruption can occur.

### Workaround

This erratum can be avoided by setting IMP_CPUACTLR_EL1[18] == 1, for example using the following code sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #18, #1
MSR S3_0_C15_C1_0, x0
```

This workaround might have a small impact on the reach of the L1 DTLB, but has an overall negligible impact on performance.

# 2314929 TLB Invalidation prevents core OFF_EMU to OFF transition

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

If a TLB invalidation instruction is executed while a core in a complex is in OFF_EMU, this instruction might prevent that core from going to OFF.

## Configurations Affected

This erratum affects all configurations with two cores in a complex.

## Conditions

The erratum occurs under the following conditions:

1. One core in the complex is in the OFF_EMU power mode.
2. The other core in the complex is in the ON, FUNC_RET, or FULL_RET power mode.
3. Either of the following is true:
    - Any core that is running executes a `TLBI` instruction.
    - The DSU receives a TLB Invalidate `SnpDVMOp` from the system interconnect.
4. The first core starts a transition to OFF, either because the PPU is reprogrammed or the debugger indicates it no longer requires OFF_EMU mode.

## Implications

If the erratum occurs, the core power transition to OFF will be denied. The core will not indicate that it needs to go to ON on its **PACTIVE** signal. Therefore it will remain in OFF_EMU and will continue to request OFF and be denied unless the core PPU is reprogrammed.

## Workaround

The system component that is programming the Power Policy Units (PPUs), typically a System Control Processor (SCP), should power the core back ON before going back to OFF_EMU and then OFF, if it detects that the core is stuck in OFF_EMU. Note that this workaround includes the workaround for errata 2275257 and 2291720 and is therefore, the same as those workarounds.

If the PPUs are being used in dynamic mode, then the following sequence will ensure that all transitions to OFF power mode are made using the OFF_EMU mode and cannot happen when the other core in the complex is powering ON:

- Set the PPU_PWPR.LOCK_EN bit if it is not already set.
- Set PPU_PWPR.PWR_POLICY to OFF_EMU instead of OFF.
- When the PPU reaches the OFF_EMU mode, the PPU_PWPR.PWR_POLICY field should be reprogrammed to OFF. Either of the LOCKED_IRQ_MASK or EMU_ACCEPT_IRQ_MASK bits in the PPU_IMR register can be cleared to request an IRQ to be generated so that the SCP knows when this mode is reached.
- The PPU_PWCR.PWR_DEVACTIVEEN field should be cleared to ensure that a new wakeup event arriving does not prevent the transition to OFF.
- The SCP should wait for the PPU to reach OFF. However, if the PPU does not reach OFF after a period of time, it might be due to the TLBI causing the transition to be denied. In this case, the following sequence should be repeated until the core reaches OFF:
    - The PPU_PWPR.PWR_POLICY field should be reprogrammed to ON.
    - Once the PPU has reached ON, the PPU_PWPR.PWR_POLICY field should be reprogrammed to OFF_EMU.
    - Once the PPU has reached OFF_EMU, the PPU_PWPR.PWR_POLICY field should be reprogrammed to OFF.
- Once the PPU has reached OFF, the PPU_PWCR.PWR_DEVACTIVEEN field can be restored to its previous value.
- The PPU_PWPR.PWR_POLICY should be set back to OFF_EMU. This can be done either as soon as the PPU has reached OFF, or it can wait until immediately before the PPU_UNLK register is written when the core is requested to power on again.
- Before the PPU_UNLK register is written to power the core on again, the SCP should check power mode of the other core in the complex. If the core is in FULL_RET, FUNC_RET, or ON then it should have its PPU_PWPR.PWR_POLICY field set to FULL_RET, FUNC_RET, or ON. Note that if the PPU_PWPR.PWR_POLICY field set to ON, then it may cause the core to leave FUNC_RET or FULL_RET if it is in either of those modes at the time, which will increase power consumption. If the field is set to FULL_RET or FUNC_RET, then if the core is ready to power off during this period, then it might temporarily transition to FULL_RET or FUNC_RET, which will increase the time taken to power OFF.
- Once the first core has reached the ON state, then the other core's PPU_PWPR.PWR_POLICY field should be restored to the value it was before the previous step.

# 2347730
# Executing a WFI/WFE with VPU powerdown enabled might result in a deadlock

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

When executing a WFI/WFE, the core may incorrectly allow the complex to enter the FUNC_RET power mode. If the core completes WFI/WFE execution during the *Vector Processing Unit* (VPU) powerdown sequence, and the instruction sequence includes a vector load, the VPU state may be corrupted, resulting in a deadlock on a later instruction.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Entry into the FUNC_RET power mode is enabled (IMP_CPUPWRCTLR_EL1.VPU_PWR_CTRL != 0b000) in all cores in the complex).
2. The program contains a WFI or WFE, followed by a vector load instruction (within the next three instructions).
3. The complex transitions into the FUNC_RET power mode, either during execution of the WFI/WFE or shortly before.
4. Precise microarchitectural timing conditions occur.

## Implications

If these conditions are met, the next FP, Advanced SIMD, or SVE instruction executed by the core might result in a deadlock.

## Workaround

This erratum can be avoided by setting CPUACTLR_EL1[17] to 1'b1, which disables specific microarchitectural clock gating behavior.

# 2358024
# EDSCR.INTdis might not mask Non-secure interrupts

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

The EDSCR.INTdis register field can mask certain interrupts if invasive debug is permitted. If Secure invasive debug is permitted, it disables all interrupts. If only Non-secure invasive debug is permitted, it should only mask interrupts taken to Non-secure.

Due to this erratum, if only Non-secure invasive debug is permitted, no interrupts will be masked.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs if all the following conditions apply:

1. DBGEN = 0b1
2. SPIDEN = 0b0
3. EDSCR.INTdis = 0b01
4. The core receives one of the following interrupts:
   - FIQ
   - IRQ
   - SError
   - Virtual FIQ
   - Virtual IRQ
   - VIrutal SError
5. The interrupt is taken to Non-secure state

## Implications

If the previous conditions are met, the core will take the interrupt when it should not.

## Workaround

The debugger can ensure all interrupts are masked by setting the following control bits:

1. PSTATE. {A, I F} == {1, 1, 1}
2. HCR_EL2. {TGE, E2H, IMO, AMO, FMO} == {0, 0, 0, 0, 0}

# 2371937
# Cacheable far atomics might generate data corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

Cacheable atomic operations issued by a core and executed beyond the L1 cache might generate data corruption.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. A core issues a cacheable atomic operation to physical address 'A'.
2. The atomic instruction is executed as a far atomic (outside the core).
3. Complex microarchitectural conditions occur.

## Implications

If the conditions are met, data corruption might happen on a different cache line. The corrupted cache line has the same set as the one targeted by the atomic operation (matching bits [X:6] of A, with X depending on L2 cache size), and potentially a different Security state.

## Workaround

Cacheable atomic operations can be forced to be executed near by setting `IMP_CPUECTLR_EL1.ATOM=0b010`, for example using the following code sequence:

```
MRS x0, S3_0_C15_C1_4
MOV x1, #2
BFI x0, x1, #38 #3
MSR S3_0_C15_C1_4, x0
```

Forcing all atomic operations to be executed near is expected to have a negligible performance impact in all systems, except in large scale systems.

# 2420992
# Incorrect instructions might be executed

## Status

Fault Type: Programmer Category B
Fault Status: Present in r1p0 and r1p1. Fixed in r1p2.

## Description

While executing conditional returns in AArch32 state, incorrect instructions might be executed and instruction cache lines might be corrupted.

## Configurations Affected

This erratum affects configurations with AARCH32_AT_EL0 parameter set to TRUE.

## Conditions

This erratum occurs under the following conditions:

1. The core executes with MMU and instruction cache is enabled.
2. The core is in AArch32 Execution state.
3. Additional complex microarchitectural conditions occur.
4. A conditional return is executed.

## Implications

If the previous conditions are met, then the core might execute incorrect instructions and might also corrupt instruction cache lines. Later execution might fetch the corrupted lines from the instruction cache, which could be at a different Exception level or within a different process. The effects of the incorrect execution are not predictable and include:

- Spurious Undefined Instruction exceptions.
- Spurious Data Aborts or other exceptions.
- Incorrect instruction execution, which might lead to register or memory state corruption.

## Workaround

Setting bit 3 of the IMP_CPUACTLR3_EL1 register (CPU Auxiliary Control Register 3) to 1 disables a power optimization feature. This will have no impact on performance, but will slightly increase power consumption (0.3-0.5%) while executing in AArch32 state.

Example code sequence:

```
MRS x0, S3_0_C15_C1_2
MOV x1, #1
BFI x0, x1, #3, #1
MSR S3_0_C15_C1_2, x0
```

# 2457168
# AMEVCNTR01 increments incorrectly

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

The AMU counter AMEVCNTR01 should increment at the same rate as the System counter, as defined by the CNT_CYCLES event. Due to this erratum, AMEVCNTR01 increments incorrectly, which gives a significantly higher result.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following condition:

1. AMEVCNTR01 is enabled to count CNT_CYCLES.

## Implications

AMEVCNTR01 cannot be used as intended to calculate the System counter frequency. The PMU is not affected because it does not implement counting of the CNT_CYCLES event.

## Workaround

Do not use AMEVCNTR01 for this purpose.

# 2658417
# BFMMLA or VMMLA instructions might produce incorrect result

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

When both cores in a complex are executing chained multiply-accumulate instructions, then under precise timing conditions a `BFMMLA` or `VMMLA` instruction might produce an incorrect result.

## Configurations Affected

This erratum affects any configuration with 2 cores in a complex, sharing a 2x64-bit VPU datapath (configuration parameter VPU_DATAPATH is set to 2x64). Configurations sharing a 2x128-bit datapath are unaffected. Affected configurations can be identified by reading IMP_CPUCFR_EL1 using `MRS Xt, S3_0_C15_C0_0` - the core is affected if bit [16] is 1 and bit [4] is 0.

## Conditions

This erratum occurs if the following conditions apply:

1. Both cores in the complex are executing chained multiply-accumulate instructions.
2. One core in the complex flushes a chained multiply-accumulate instruction executed speculatively.
3. A `BFMMLA`, `VMMLA` or multicycle vector instruction is executed after the flush.
4. Precise microarchitectural timing conditions occur.

Any of the following instructions are classed as chained multiply-accumulate instructions:

- `BFMMLA`
- `BFDOT`
- `VMMLA`
- `VDOT`
- `VMLA/VMLS`
- `VNMLA/VNMLS`
- `VRECPS`
- `VRSQRTS`

Any of the following instructions are classed as multicycle vector instructions:

- `FDIV*`

- `FSQRT`
- `SDIV*/UDIV*`
- `BDEP/BEXT/BGRP`
- `PMULL*`
- `RAX1`
- `SHA1C/SHA1M/SHA1P`
- `SHA256*/SHA512*`
- `SM3*/SM4E*`
- `VDIV`
- `VMULL`
- `VSQRT`

## Implications

If these conditions are met, the result of a **`BFMMLA`** or **`VMMLA`** instruction executed by either core might be incorrect. As FEAT_BF16 and FEAT_AA32BF16 are recent additions to the architecture, these instructions are not expected to be present in the legacy code.

## Workaround

There is no complete workaround for this erratum. It is expected that software will use run-time feature detection to determine whether to use these instructions or to fall back on support for earlier architecture versions. A kernel can avoid this erratum by updating the detected feature list to remove FEAT_BF16 and FEAT_AA32BF16 from the list of supported features in affected systems.

# 2666669
# Data corruption might occur during core powerdown

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

Data corruption might occur when a core is powered down.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. The core executes a PRFM PLDL1 or PRFM PSTL1 shortly before a WFI.
2. The core is requested to power off.
3. Timing sensitive microarchitectural conditions occur.

## Implications

If the conditions are met, dirty data might be lost on the cache line accessed by the PRFM instructions, resulting in data corruption.

## Workaround

To prevent this erratum from occurring, software can set IMP_CPUACTLR_EL1[38] = 1, for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #38, #1
MSR S3_0_C15_C1_0, x0
```

This might impact the effectiveness of some PRFM instructions. This is unlikely to have a measurable performance impact.

## 2675636
## An asynchronous MTE check might not observe correct memory ordering

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

### Description

An asynchronous *Memory Tagging Extension* (MTE) check on a store might not be correctly ordered with respect to an earlier DMB or instruction with acquire semantics.

### Configurations Affected

This erratum affects configurations with BROADCASTMTE=true.

### Conditions

This erratum occurs under the following conditions:

1. MTE checking is enabled, and set to accumulate faults asynchronously for writes (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b1x).
2. The core executes an instruction with acquire semantics, or a DMBLD/DMBSY.
3. A checked store instruction is executed to Inner Writeback and Outer Writeback, tagged, memory, and does not have an address dependency w.r.t. an ordered-before load.
4. Timing-sensitive, micro-architectural conditions occur.

### Implications

If the conditions are met, the MTE check on the store might not be correctly ordered w.r.t. the DMB or instruction with acquire semantics. This might result in an incorrect update of the TFSR_ELx register.
If the checked store instruction has an address dependency on a load ordered after the instruction at condition (2), or the instruction at (2) itself, the erratum does not occur.

### Workaround

Arm believes common memory allocator code is likely to have such an address dependency, and as a result does not require a workaround. If a workaround is required, software can avoid enabling asynchronous MTE checking for store operation, by setting SCTLR_ELx.TCFn = 0b01.

# 2684597
# A core might deadlock during powerdown if TRBE is enabled

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

## Description

A core might deadlock during powerdown if *TRace Buffer Extension* (TRBE) is enabled.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The TRBE is enabled by setting TRBLIMITR_EL1.E = 0b1. The core is executing in a non-prohibited trace region.
2. The core executes a WFI, WFIT, WFE or WFET instuction.
3. An external wakeup or timeout occurs, waking up the core.
4. The core executes a WFI instruction to power down the core.
5. system rely on the core to be ready to enter to power OFF/OFF_EMU state
6. Timing-sensitive micro-architectural conditions occur.

## Implications

If the conditions are met, the core might not accept the power state transition to OFF/OFF_EMU state and system may deadlock.

## Workaround

Software can execute a `TSB CSYNC` and `DSB before execute WFI for power down`.

# 2724177 Deferred error might become uncontainable

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, and r1p2. Fixed in r1p3.

## Description

Poison information cached in a *Processing Element* (PE) might be lost and therefore make the deferred error to become uncontainable.

## Configurations Affected

This erratum affects configurations having all of the following:

1. Parameter CORE_CACHE_PROTECTION is TRUE.
2. The interconnect does not have a precise snoop filter, and does not use SnpQuery to inquire about the state of the line at the *Request Node* (RN).

## Conditions

1. A line is cached in the PE and in another cache in the system.
2. A MakeReadUnique from the complex is processed by the interconnect and poisoned data is returned, without the line being lost by the complex.

## Implications

If the condition occurs, the line might be propagated to the core without being poisoned.

## Workaround

Software can enable the Error Recovery Interrupt for deferred error by setting the DUI bit of all the *Reliability*, *Availability*, *and Serviceability* (RAS) node registers ERRxCTLR to prevent the error from becoming uncontainable.

# 2971420 Data corruption or deadlock might happen if TRBE is enabled

## Status

Fault type: Programmer Category B
Fault status: Present in r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, and r1p3. Open.

## Description

Data might be corrupted if *Trace Buffer Extension* (TRBE) is enabled.

## Configurations affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The TRBE is enabled by setting TRBLIMITR_EL1.E = 0b1.
2. The trace data store to the memory cross 4k page.
3. One of the 4k page has translation fault or permission fault.
4. Timing-sensitive micro-architectural conditions occur.

## Implications

If the above conditions are met, random data might be corrupted or deadlock might happen before the core takes the TRBE IRQ.

## Workaround

The EL3 firmware can prevent trace collection via TRBE by programming MDCR_EL3.NSTB[1] to the opposite value of SCR_EL3.NS on a security state switch.

# 3057514
# Deferred error might become uncontainable when BROADCASTMTE is set

## Status: Open

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, and r1p3. Open.

## Description

Poison information cached in a *Processing Element* (PE) might be lost and therefore make the deferred error to become uncontainable.

## Configurations affected

This erratum affects configurations having parameter CORE_CACHE_PROTECTION set to TRUE, and the parameter BROADCASTMTE set to TRUE.

## Conditions

This erratum might occur if line A (Normal Inner Write-Back, Outer Write-Back Cacheable) is cached in the PE and another cache in the system, one cache has the line as poisoned, and the following condition is met:

- The non-L1 allocating store operation executed by this PE and the store operation modifies less than a cache line worth of MTE tags.

## Implications

If the condition occurs, the line might be propagated to the core without being poisoned.

## Workaround

Software can enable the Error Recovery Interrupt for deferred error by setting the DUI bit of all the *Reliability, Availability, and Serviceability* (RAS) node registers ERRxCTLR, and treat all deferred errors as uncontainable.

# 3117295
# A speculatively executed unprivileged load might leak data from a privileged via side channel

## Status

Fault Type: Programmer Category B.
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3. Open.

## Description

A speculatively executed unprivileged load might leak data from a privileged via side channel.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. A load is speculatively executed at EL0, accessing a location in memory that is mapped, but lacks EL0 access permissions.
2. Timing-sensitive, microarchitectural conditions occur.

## Implications

The speculatively executed load can, under specific microarchitectural conditions, speculatively forward data, bypassing a permission check, to the address operand of another load, potentially leaking information from a higher privilege level via side channel.

Pagetable isolation between EL0 and higher level ELs prevents the issue from occurring.

## Workaround

If pagetable isolation is disabled, the context switch logic in the kernel can be updated to execute the following sequence on affected cores before exiting to EL0, and after all explicit memory accesses:

1. A non-shareable TLBI to any context and/or address, including unused contexts or addresses, such as a `TLBI VALE1 Xzr`.
2. A DSB NSH to guarantee completion of the TLBI.

# Category B (rare)

## 1976290
## A Tag Checked load might forward incorrect data

### Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

A Tag Checked load might forward data to dependent instructions that is inconsistent with the data written to the architectural register file.

### Configurations Affected

This erratum affects all configurations where the **BROADCASTMTE** pin is HIGH.

### Conditions

The erratum occurs under the following conditions:

1. MTE checking of loads and stores is enabled.
2. The core executes a store to a Normal Inner Write-back, Outer Write-Back Cacheable location.
3. The core executes a load to the same cache line as the cacheable store above.
4. The core executes a Tag Checked load to the same location as the load above, and the load hits in the L0 cache.
5. Another PE in the system modifies the location loaded by the loads above, concurrently.
6. Rare, timing-sensitive microarchitectural conditions occur.

### Implications

If the conditions are met, the last load might forward data to dependent instructions that is inconsistent with the data written to the architectural register file.

### Workaround

This erratum can be avoided by setting IMP_CPUACTLR_EL1[4] = 1; for example, using the following code sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
```

```
BFI x0, x1, #4, #1
MSR S3_0_C15_C1_0, x0
```

This workaround disables early load data return and might have a measurable performance impact.

# 2002389
# Asynchronous exceptions after ECC errors might cause unpredictable behavior

## Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

When an unmasked interrupt is pending during a specific sequence of load/store instructions and branch instructions, and under complex and rare microarchitectural conditions that require an ECC error (single bit, double bit or poison), unpredictable behavior might be observed.

## Configurations Affected

All configurations are affected.

## Conditions

1. The following sequence of instructions is executed:
    a. Load/Store that generates an ECC error.
    b. Branch instruction
    c. Load/Store that generates an ECC error
2. Either one of the following:
    - The instruction executed in step 1.1 is one of the following types:
        - SVE load multiple structures
        - SVE contiguous load
        - SVE contiguous non-fault load
        - SVE contiguous first-fault load
        - SVE contiguous non-temporal load
        - SVE Gather loads (all variants)
        - LDx (multiple structures) - All variants
        - LDx (single structures) - All variants
        - `LD3R`
        - `LD4R`
        - Atomic memory operations - All variants
    - Embedded Trace Macrocell (ETM) tracing is enabled (TRCPRGCTLR.EN == 1)
3. An interrupt is pending and is unmasked according to the table in section "Asynchronous exception masking" in the Armv8-A Architecture Reference Manual.
4. Complex and rare microarchitectural conditions are met.

## Implications

If these conditions are met, then unpredictable behavior in the core might occur, including execution of incorrect instructions or at an incorrect Exception level. This includes, but is not limited to, incorrect entry to EL0 and/or incorrect switching to Non-secure state.

PSTATE.IL will always be set when these conditions are met.

Security state escalation is not possible when these conditions are met.

## Workaround

If workaround is required this erratum can be avoided by setting IMP_CPUACTLR_EL1[3] = 0b1; for example, using the following code sequence:

```
MRS X0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #3, #1
MSR S3_0_C15_C1_0, x0
```

This is expected to impact performance by around 1.3%.

# 2080326
# A longer TLBI sequence might cause a deadlock

## Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

Under rare conditions, a longer sequence of TLBI instructions might cause a deadlock.

## Configurations Affected

This erratum affects configurations with two cores in a complex.

## Conditions

The erratum occurs under the following conditions:

1. At least one core in the complex has 10 or more TLBI outstanding, without intervening DSBs.
2. Rare, timing-sensitive micro-architectural conditions occur.

## Implications

If the conditions are met, a deadlock can occur.

## Workaround

From the r0p2 version, it is possible to prevent this erratum from occurring by executing the following instruction sequence:

```
MOV     x0, #1
MSR     S3_6_C15_C4_0, x0
ISB

MOV     x0, #0x0100
MOVK    x0, #0x0E08, LSL #16
MSR     S3_6_C15_C4_2, x0

MOV     x0, #0x0300
MOVK    x0, #0x0F1F, LSL #16
MOVK    x0, #0x0008, LSL #32
MSR     S3_6_C15_C4_3, x0

MOV     x0, #0x03F1
MOVK    x0, #0x00C0, LSL #16
MSR     S3_6_C15_C4_1, x0
```

```
ISB
```

This will perform a Data Synchronization Barrier after each **TLBI** instruction. This is expected to have a minimal performance impact.

# 2277097
# CMOs to non-shareable locations might deadlock the cluster

## Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

A data cache maintenance operation to a non-shareable location might cause a deadlock if unlikely timing conditions are met.

## Configurations Affected

Only systems with interconnects that can send a CompCMO for a combined WriteNoSnoop+CMO before receiving the write data are affected.

## Conditions

1. A data cache maintenance instruction by address is issued.
2. Target PA is marked as non-shareable and is dirty.
3. Interconnect issues a CompCMO before the data is received, and this response arrives together CompDBIDResp to the CPU core.

## Implications

If the above conditions are met, the CPU core might deadlock. Most system interconnects will respond with CompCMO only after having received the data, so this errata does not apply to those kind of systems.
Since the DSU issues CompDBIDResp to CPU at the same time that the combined Write+CMO is sent to the interconnect, Arm believes it to be very rare for a CPU core to receive both the CompDBIDResp and CompCMO at the same time.

## Workaround

In order to avoid this erratum, non-shareable memory should not be used.

# 2441009
# Completion of affected memory accesses might not be guaranteed by completion of a TLBI

## Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

The core might not guarantee completion of all memory accesses after completion of a TLB Invalidate (`TLBI`) instruction affecting those accesses on another core.

## Configurations affected

This erratum affects all configurations.

## Conditions

1. Another PE in the system executes a `TLBI` or Instruction Cache (`IC`) instruction, followed by a Data Synchronization Barrier (`DSB`) instruction.
2. The core executes a store to a memory location A.
3. Another PE in the system modifies the descriptor used by the store to memory location A, using a break-before-make sequence.
    - The break-before-make sequence will include a `TLBI` instruction, followed by a `DSB` instruction.
4. Rare, timing-sensitive, microarchitectural conditions occur.

## Implications

The `DSB` used after the `TLBI` as part of the break-before-make sequence might not guarantee the completion of the store to memory location A under very rare and unlikely timing conditions. For most systems and applications, the latency of the break-before-make sequence and time until later reuse is very likely to exceed the latency required to naturally complete the store.

## Workaround

Given the rarity of the conditions needed to trigger this erratum, a workaround is not expected to be needed in most systems.
If a workaround is required, then the `TLBI`, `DSB` sequence from the break-before-make sequence can be repeated. After repeating the `TLBI`, `DSB` sequence, all memory accesses that use a translation changed by the break-before-make sequence will have completed.

# Category C

## 1898949
## Execution of an atomic instruction may fail to make forward progress

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

Execution of an atomic instruction that is followed by a (speculative) load to the same cache line within the next 5 memory instructions (loads, stores, and atomics) may hang if another PE in the system executes a steady sequence of stores or atomics to the same cache line.

### Configurations Affected

All configurations are affected.

### Conditions

1. The core executes an atomic instruction to inner-writeback/outer-writeback memory. This atomic is performed "near", in the core itself.
2. The core (speculatively) executes a load instruction to the same PA.
3. The cache line is actively being written to by other PEs in the system in an uninterrupted sequence.

### Implications

This erratum can allow denial of service between two threads that share memory.

### Workaround

It is not expected that software running on LAC-quality silicon samples will encounter this erratum. No workaround available.

# 1904476 Reads of GMID_EL1 might trap to incorrect exception level when MTE is disabled

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

When the *Memory Tagging Extension* (MTE) is disabled and the HCR_EL2.TID5 trap is enabled, reads of the GMID_EL1 register at EL1 are incorrectly trapped to EL2 instead of taking an undefined exception at EL1.

## Configurations Affected

This erratum affects configurations where the **BROADCASTMTE** pin is LOW.

## Conditions

- HCR_EL1.TID5 is set to 1.
- The core is executing at EL1.
- An `MRS` instruction from GMID_EL1 is executed.

## Implications

If the conditions are met, then the trap on the read of GMID_EL1 will be taken to EL2 instead of EL1.

## Workaround

When MTE is disabled (ID_AA64PFR1_EL1.MTE reads as 0x1), the erratum can be avoided by not setting HCR_EL2.TID5.

# 1905896
# Error responses to MakeReadUnique transactions might result in data corruption

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If an error response is returned for a MakeReadUnique transaction, the core may load corrupt data for that cache line.

## Configurations Affected

Configurations of the cluster using a CHI interface (configuration parameters CHI or PERIPH_PORT_CHI are TRUE) are affected.

## Conditions

1. The core has a cache line A in the L1 cache.
2. The cluster holds the cache line in a Shared coherency state.
3. The core executes a store to the cache line.
4. A MakeReadUnique transaction to the cache line is sent from the cluster.
5. No other coherent agents write to the cache line.
6. The response to the MakeReadUnique transaction has an error status (DErr or NDErr).

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. If the conditions are met, subsequent loads of the cache line executed on the core might return corrupt data, even after a subsequent write to the cache line.

## Workaround

No workaround is required.

# 1911617
# Tag Check Fault reporting might be incorrect when ECC error occurs

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

An ECC error in the L1 D-cache MTE data RAM might cause the core to incorrectly report a Tag Check Fault, or fail to report a Tag Check Fault.

## Configurations Affected

This erratum affects configurations where the **BROADCASTMTE** pin is tied HIGH.

## Conditions

1. SCTLR_ELx.TCF is set to 0b10, or SCTLR_ELx.TCF0 is set to 0b10.
2. ERR1CTLR.ED is set to 1.
3. The core is in Write Streaming mode.
4. A Tag Checked store is executed.
5. The store hits in the L1 D-cache.
6. There is a correctable error in the MTE data RAM when the L1 D-cache is read.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. When these conditions are met, the core might report a spurious Tag Check Fail or fail to report a Tag Check Fail. The error in the MTE data RAM will be correctly reported in the RAS error record registers.

## Workaround

No workaround is required.

# 1915215
# Software stepping ERETAA or ERETAB might set SPSR.ELx.SS to incorrect value

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A pointer authentication failure when software stepping **ERETAA** or **ERETAB** in the active-not-pending state, fails to set SPSR_ELx.SS when PSTATE.BTYPE does not equal 0.

## Conditions

1. SPSR_ELx.BTYPE is non-zero.
2. An exception return is then executed, which enables software stepping (entering the active-not-pending state).
3. **ERETAA** or **ERETAB** is executed outside of a guarded page, and the associated PAC operation fails.

In the first condition, SPSR_ELx.BTYPE will typically be non-zero if the previously stepped instruction before the **ERETAA** or **ERETAB** was a **BR**, **BRAA**, **BRAAZ**, **BRAB**, **BRABZ**, **BLR**, **BLRAA**, **BLRAAZ**, **BLRAB**, or **BLRABZ** instruction.

## Implications

If the conditions are met, then the value of SPSR_ELx.SS after taking the PAC exception will be 0 rather than 1. This is unlikely to cause issues for most software because the PAC exception will be treated as fatal by the handling software, so the value of SPSR_ELx.SS in this case will be unused.

## Workaround

There is no workaround.

# 1919823
# Configuration of IMP_CMPXECTLR_EL1.CDEVC not supported

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Setting IMP_CMPXECTLR_EL1.CDEVC, bits [45:44], to a value other than 0b11 might cause deadlock. Note that 0b11 is the reset value for this register.

## Configurations Affected

All configurations are affected.

## Conditions

1. The value of IMP_CMPXECTLR_EL1.CDEVC, bits [45:44], is not 0b11.
2. A core in a complex executes a **`DC CVAP`** or **`DC CVADP`** instruction for a line that is cached clean within the complex.

## Implications

If these conditions are met, then one or more cores, or other coherent agents in the system might deadlock.

## Workaround

To avoid this erratum, ensure that IMP_CMPXECTLR_EL1[45:44] is set to 0b11. For example, use the following code sequence:

```
MRS x0, S3_0_C15_C1_7
MOV x1, #3
BFI x0, x1, #44, #2
MSR S3_0_C15_C1_7, x0
```

## 1921977
## Error response to load-exclusive might cause loss of synchronization or deadlock

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If a Load-Exclusive instruction observes an external memory error, a subsequent Store-Exclusive instruction might deadlock or incorrectly update memory.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. A Load-Exclusive instruction is executed to Shareable and Cacheable memory
2. The access misses in the cluster caches and sends a transaction on the external interface
3. The data returned by the external transaction indicates an error (DErr on CHI, or SLVERR/DECERR on AXI), but only on data beats not containing the data loaded by the Load-Exclusive instruction
4. A Store-Exclusive instruction is executed to the same cache line, without an intervening `CLREX` instruction or store to the same cache line

### Implications

If the Store-Exclusive is Tag Checked, the core might deadlock.

After the Load-Exclusive instruction, the global monitor will be in the Exclusive Access state, but might not move into the Open Access state if there is a store to the marked cache line from another PE. Subsequently, the Store-Exclusive updates memory despite another store to the location having occurred.

### Workaround

There is no workaround.

# 1924991
# Accesses to TRCQCTLR are RAZ/WI instead of UNDEFINED

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The core does not implement support for Q elements as part of the Embedded Trace Extension. As such, the TRCQCTLR register is unimplemented. The reads or writes of the TRCQCTLR register behave as if it is implemented as RAZ/WI, instead of being UNDEFINED, because of this erratum.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is executing at EL1 or above.
2. An **`MSR`** or **`MRS`** instruction to TRCQCTLR is executed.

## Implications

Accesses to TRCQCTLR will be trapped (similar to accesses to implemented trace registers) because of the CPACR_EL1.TTA and CPTR_ELx.TTA bits. If not trapped, accesses to TRCQCTLR are RAZ/WI.

## Workaround

There is no workaround.

## 1925280
## A single-bit error in the L2DB RAMs might be reported incorrectly as a corrected error

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

Some single-bit errors detected in the L2 data buffer (L2DB) RAMs are not corrected but are incorrectly reported as corrected.

### Configurations Affected

This erratum affects configurations with core cache protection enabled.

### Conditions

1. A single-bit error is detected in the L2DB RAMs on an L2 cache allocation.

### Implications

If the conditions are met, the error is not corrected and is instead deferred, but is incorrectly reported as corrected in ERR2STATUS.CE. It is expected the error will be corrected on a later read from the L2 data RAMs.

### Workaround

There is no workaround.

# 1929084
# Direct access to L1 data cache MTE data RAMs does not function

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The core provides system register encodings to allow the contents of the L1 data cache to be read directly from EL3. The contents of the memory containing the Memory Tagging Extension (MTE) tags cannot be read.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. A read of the L1 D-cache MTE RAMs is initiated using `SYS #6, C15, C2, #4, <Xt>`

## Implications

A subsequent read of IMP_CDBGDR0_EL3 will return zero. The contents of the MTE tag memories in the L1 data cache cannot be read directly.

## Workaround

There is no workaround.

# 1933869
# Writes to DBGWCR<n>_EL1/DBGWVR<n>_EL1 might cause speculative reads of device memory

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

An external APB access or software `MSR` writing to DBGWCR<n>_EL1 or DBGWVR<n>_EL1 during the watchpoint check of a stalled instruction might cause both an abort and an access to device memory.

## Conditions

1. DBGWCR<n>_EL1 or DBGWVR<n>_EL1 is written, either using an `MSR` instruction without a subsequent `ISB` or `DSB`, or via the Advanced Peripheral Bus (APB) interface.
2. At around the same time, a load instruction is executed to device memory that generates a watchpoint debug event because of one of the watchpoint register pairs that had been written.

## Implications

If the conditions are met, then the load instruction will take an exception or cause a Debug state entry, but a read transaction will still be generated to the address targeted by the load. If this address maps to a read-sensitive peripheral, the side effects of the read will be observable despite the load not having retired. This erratum does not affect the programming of watchpoints for inactive contexts.

## Workaround

Debuggers should avoid setting watchpoints on read-sensitive locations. An `ISB` or `DSB`, when inserted after an `MSR` of the watchpoint registers and before the load, guarantees that the load will observe only the updated value and prevents a speculative access to device memory.

# 1934328
# Multiple errors detected on the same cycle might lead to an incorrect value of ERR2STATUS and ERR2MISC0

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

If multiple errors are detected on the same cycle across the L2 tag, L2 data and L2 data buffer (L2DB) RAMs, the value recorded in ERR2STATUS and ERR2MISC0 might be incorrect.

## Configurations Affected

This erratum affects configurations with core cache protection enabled.

## Conditions

1. Multiple errors are detected on the same cycle across the L2 tag, L2 data and L2DB RAMs.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. Arm believes that detection of multiple errors on the same cycle is exceedingly rare. If the conditions are met, the value recorded in ERR2STATUS and in ERR2MISC0 might not correctly reflect all detected errors, but only a subset or a mix of these.

## Workaround

There is no workaround.

# 1946400
# Error reporting disabled does not mask error responses on memory accesses

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Error responses on memory accesses, including speculative memory accesses, might be reported in the RAS registers when ERR1CTLR.ED = 0.

## Configurations Affected

All configurations are affected.

## Conditions

1. ERR1CTLR.ED = 0.
2. A memory access by a load or store instruction, or a speculative L1 data cache refill sees an error or a poison value on the response.

## Implications

If these conditions are met, an uncorrected error is erroneously reported.

- ERR1STATUS.V = 1.
- ERR1STATUS.UE = 1 and ERR1STATUS.ER = 1.
- If the error was generated as the result of data poison, ERR1STATUS.PN = 1.
- ERR1STATUS.SERR = 0x18 for NDerr and ERR1STATUS.SERR = 0x12 for Derr and data poison.

## Workaround

There is no workaround for this erratum.

# 1951345
# PMU_HOVFS event exported when EL2 trace disabled

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

PMU_HOVFS is a PMU event that can be exported to the ETM. Exporting this event is disabled if TRFCR_EL2.E2TRE == 0b0. Due to this erratum, the event is exported regardless of this setting.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The ETM is configured to use PMU_HOVFS as an external input event.

## Implications

Overflows of PMU counters reserved by EL2 might be visible to lesser-privileged software.

## Workaround

There is no workaround.

## 1951423
## An ECC error during core power down might cause another error to occur after power up

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If an ECC error is corrected in the L1 data cache duplicate tags for a cache line that is invalid, an incorrect value might be written into the L1 data cache duplicate tag RAMs. This can occur after a power off/power on cycle during the hardware invalidation of the affected L1 data cache duplicate tags.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. An error is detected, corrected and reported on the L1 data cache duplicate tag RAMs.
2. The core corresponding to the above L1 data cache duplicate tag RAM is powered down, and no cache line is flushed after the above error detection (for example, because the L1 data cache was already empty, or the error was detected during the hardware flush).
3. The same core is powered on again.

### Implications

If the above conditions are met, the automatic hardware-based invalidation of the duplicate tag RAMs during core power-up might write an incorrect value into the RAM. This might result in a spurious error report, either correctable or uncontainable.

### Workaround

There is no workaround.

# 1951568
# PMU event counts might be inaccurate

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The processor implements a number of performance monitor events. Because of this erratum, some of the events will not count correctly.

## Configurations Affected

All configurations are affected.

## Conditions

1. One of the PMU event counters is configured to count one of the following events:
    - 0x0006 LD_RETIRED
    - 0x001C TTBR_WRITE_RETIRED
    - 0x0029 L3D_CACHE_ALLOCATE
    - 0x0037 LL_CACHE_MISS_RD, if IMP_CPUECTLR_EL1.EXTLLC is not set
    - 0x00A2 L3D_CACHE_REFILL_RD
    - 0x00C8 LL_WS_MODE
    - 0x00EE STALL_SLOT_BACKEND_ILOCK

## Implications

Software will not be able to use the above events for performance analysis.

## Workaround

There is no workaround.

# 1954191
# L2 cache debug operations might hang and block a power off request

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If an L2 cache debug operation is performed during the power down of a core in the complex, or the complex itself, both the cache debug and the power transition might hang.

## Configurations Affected

This erratum affects configurations with two cores in the complex, and where the complex is configured with an L2 cache (parameter L2_CACHE is TRUE).

## Conditions

1. A core in the complex is in the process of powering off.
2. A cache debug operation to the L2 cache (**`SYS IMP_CDBGL2CDR`**, **`SYS IMP_CDBGL2CMR`** or **`SYS IMP_CDBGL2CTR`**) is performed by the other core in the complex during the above power-off operation.

## Implications

Both the power-off, and the cache debug operation might hang. The power-off request hang might result in a P-channel hang.

## Workaround

Software should avoid cache debug operations when one of the cores might potentially be powered off.

# 1954658
# ERR2STATUS.CE might be incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If an uncorrected, deferred, or uncontainable error is detected after a corrected error was detected, ERR2STATUS.CE might be incorrect.

## Configurations Affected

This erratum affects configurations with core cache protection enabled.

## Conditions

1. A corrected error is detected in the L2 tag RAMs, L2 data RAMs or L2DB RAMs.
2. An uncorrected, deferred, or uncontainable error is detected in the L2 tag RAMs, L2 data RAMs or L2 data buffer (L2DB) RAMs.

## Implications

If the conditions are met, ERR2STATUS.CE might not reflect that a corrected error was detected. ERR2MISC0.CECO remains correct and can be used for statistical purposes.

## Workaround

There is no workaround.

# 1955046
# External events are not observed after Warm reset

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description:

External events can be programmed to be observed by the Embedded Trace Extension (ETE) and are observed if ETE is not in low power state or disabled. Because of this erratum, under some conditions the external events are not observable by ETE.

## Conditions

1. The Embedded Trace Extension unit is enabled.
2. At least one of TRCEXTINSELR<n> is programmed to trace external events.
3. Warm reset is applied and released.
4. An external event occurs.
5. Exact microarchitectural timing conditions occur.

## Implications

If these conditions are met, the external events are not observed by ETE.

## Workaround

There is no workaround.

# 1955072
# Mapping memory as Tagged when Allocation Tag storage is not provided might lead to CHI protocol violations

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If a page is mapped as Tagged, but the underlying tag physical address (PA) does not provide Allocation Tag storage, direct and indirect accesses to the Allocation Tags in that page may lead to violations of the CHI protocol.

## Configurations Affected

This erratum affects all configurations where the **BROADCASTMTE** pin is HIGH.

## Conditions

1. A page is mapped as Tagged, but Allocation Tag storage for that tag PA is not provided.
2. A direct or indirect access to Allocation Tags within that page occurs.

## Implications

The direct or indirect access to Allocation Tags in the page may cause violations of the CHI protocol.

## Workaround

Software should not map pages which do not provide Allocation Tag storage as Tagged.

# 1956538
# Execution of ESB instruction in debug state might cause unpredictable behavior

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The execution of an **ESB** instruction while in the debug state might result in unpredictable behavior in the core to be observed, if an SError is pending.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core enters the debug state.
2. The external debugger passes an **ESB** instruction to the core via the EDITR.
3. The core executes the **ESB**.
4. A virtual or physical SError is pending, such that the **ESB** instruction updates DISR_EL1 and/or VDISR_EL2.

## Implications

If the above conditions are met, then unpredictable behavior in the core might occur, including execution of incorrect instructions or at an incorrect Exception level. This includes, but is not limited to, incorrect updates to the DLR and DSPSR.

Privilege/security state escalation is not possible when these conditions are met.

## Workaround

The Debugger should save/restore the DLR and DSPSR when executing an **ESB** instruction.

# 1959615
# Exceptions in debug state might allow execution of subsequent instruction in EDITR

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

An exception generated in debug state might allow further issue of a subsequent instruction from EDITR (when EDSCR.ERR == 1).

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core enters debug state.
2. The external debugger passes an instruction which might generate an exception (such as a data abort) to the core via the EDITR.
3. On execution of the instruction from step 2, a synchronous exception is generated.
4. Microarchitectural timing conditions occur.

## Implications

If the above conditions are met, then an instruction in EDITR might be executed when EDSCR.ERR == 1.

## Workaround

External debugger can poll EDSCR.ITE prior to writes to EDITR.

# 1964078
# VMID tracing incorrectly allowed or disallowed based on TRFCR_EL2.CX

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The VMID field of a trace packet may be incorrectly masked or unmasked compared to the value of TRFCR_EL2.CX.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. ETM tracing is enabled (TRCPRGCTLR.EN == 1).
2. Self hosted trace is enabled.
3. CID_EL2 and VMID trace are allowed (TRCCONFIGR.VMID == 1 AND TRFCR_EL2.CX == 1).
4. Core is executing in Secure EL0/EL1 with Secure EL2 disabled (SCR_EL3.NS == 0 AND SCR_EL3.EEL2 == 0).

OR:

1. ETM tracing is enabled (TRCPRGCTLR.EN == 1).
2. A P0 element executes immediately prior to an MSR that updates TRFCR_EL2.CX.

## Implications

If the first set of conditions are met, then the VMID field in trace elements might be zero, potentially leading to the generation of incorrect context elements by the trace unit.

If the second set of conditions are met, then trace elements might have the incorrect VMID associated with them, potentially leading to the generation of incorrect context elements by the trace unit.

## Workaround

No workaround is required for the first set of conditions in this erratum, as the architecture is expecting that EL3 will clear TRFCR_EL2 as part of a state switch in these conditions.

No workaround is required for the second set of conditions, as software is not expected to update TRFCR_EL2.CX when tracing is enabled.

# 1965154
# PMU snapshot records incorrect Context ID when PMU inactive

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The PMU snapshot feature enables sampling of various parts of the execution state. Because of this erratum, Context ID values may not be sampled correctly while the PMU is inactive.

## Configurations Affected

All configurations are affected.

## Conditions

The erratum occurs under the following conditions:

1. All PMU counters are disabled by setting MDCR_EL2.HPME == 0 and PMCR_EL0.EN == 0.
2. PMU event exporting is disabled by ETM trace being disabled.
3. A PMU snapshot is requested.

## Implications

The Context ID values in a PMU snapshot can be incorrect

## Workaround

If any PMU counters or ETM trace are enabled while a snapshot is taken, Context ID will be sampled correctly.

# 1966395
# A continuous stream of stores might prevent forward progress of a coherency operation

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A continuous stream of stores on a core might prevent forward progress of a coherency operation, leading to a denial of service and perceived hang on a different PE or component in the system.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. A continuous stream of non-L1-allocating stores (including stores to device and non-cacheable memory), `DC ZVA`, `DC GVA` or `DC GZVA` is executed on a core.
2. A coherency operation is received for a cache line present in the L1 cache of that core.

## Implications

If the conditions are met, the coherency operation might not make forward progress while the store stream is ongoing.

## Workaround

A denial of service on a secure process can be avoided if the secure OS sets up a timer-based interrupt source to interrupt all cores periodically.

# 1974927
# SIMD, FP, and SVE cacheable loads or MTE-checked stores might hang on accessing a poisoned cache line

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

SIMD, FP, and SVE loads or MTE-checked stores to cacheable memory might hang if at least one cache line being accessed is poisoned due to an earlier deferred error.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under one of the following two sets of conditions:

Set 1:

1. The core executes a SIMD, FP, or SVE load to Normal Inner Write-Back, Outer Write-Back Cacheable memory. The load is one of the following types:
   a. SVE load multiple structures
   b. SVE contiguous load
   c. SVE contiguous non-fault load
   d. SVE contiguous first-fault load
   e. SVE contiguous non-temporal load
   f. SVE Gather loads (all variants)
   g. LDx (multiple structures) - All variants
   h. LDx (single structures) - All variants
   i. **`LD3R`**
   j. **`LD4R`**
2. At least one cache line accessed by the load is poisoned due to a deferred error.

Set 2:

1. MTE checking of stores is enabled, and set to generate synchronous exceptions on tag check fails (SCTLR_ELx.TCF = 0b01).
2. The core executes an MTE-checked SIMD, FP, or SVE store to Normal Inner Write-Back, Outer Write-Back Cacheable, Tagged memory. The store is one of the following types:

   a. SVE scatter/contiguous store
   b. STx (store multiple) - All variants
3. At least one cache line accessed by the store is poisoned due to a deferred error.

## Implications

If the conditions are met, the load or MTE-checked store might hang, preventing further execution until the poison is cleared. Interrupts will not be taken. This is not expected to be a significant issue for sample silicon.

## Workaround

There is no workaround.

# 1975007
# WFE trap might take effect when a wake up event is pending

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A trap (as configured by SCTLR_ELX.nTWE, HCR_EL2.TWE, SCR_EL3.TWE) might be taken upon execution of a **`WFE`** instruction that would not otherwise cause entry to low power state due to a pending, unmasked interrupt.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Entry to WFE low-power state is configured to be trapped upon execution in current exception level (trap configured such that ESR_ELx.EC would == 0b000001).
2. No WFE wake up events are pending.
3. An architectural event unmasks interrupt type <X> as per table D1-13 of the Arm ARM (for physical interrupts) or table D1-17 of the Arm ARM (for virtual interrupts).
4. A **`WFE`** instruction is executed whilst an interrupt of type <X> is pending.

## Implications

If these conditions are met, then execution of the **`WFE`** instruction will be trapped when there is an architecturally defined wake up event pending.

## Workaround

There is no workaround.

# 1976399
# PrefetchTgt transactions are generated when PrefetchTgt is disabled

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

PrefetchTgt transactions are generated even when PrefetchTgt is disabled through control bits in IMP_CMPXECTLR_EL1[39:38]. The default setting of these control bits is to not generate PrefetchTgt transactions.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. A memory access instruction (including speculative execution), to Normal Inner Write-Back, Outer Write-Back Cacheable memory results in an L1 and L2 miss.
2. PrefetchTgt transactions for demand data memory accesses are disabled with IMP_CMPXECTLR_EL1[39:38] = 0b00.

## Implications

If the conditions are met, a PrefetchTgt transaction might be generated even though the control bits disable their use. The generation applies to a subset of demand data memory accesses.
The unexpected generation of PrefetchTgt transactions should not cause functional issues in a system that has support for PrefetchTgt transactions.

## Workaround

In a system where the use of PrefetchTgt transactions must be fully disabled, the software can set IMP_CPUACTLR_EL1[14] = 1; for example, by using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #14, #1
MSR S3_0_C15_C1_0, x0
```

This is not expected to have a material performance impact in common use cases, but it does increase the best-case L1 data miss latency by one cycle.

# 1977191
# Trigger received immediately out of warm reset might cause deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

When a PE comes out of reset, it performs some internal handshaking to obtain certain configuration parameters before it can start execution. Due to this erratum, if an Embedded Logic Analyzer (ELA) or Embedded Trace Macrocell (ETM) trigger is received by the PE just as it starts this handshake, the handshake might never complete and the PE cannot come out of reset correctly.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The PE receives an ELA or ETM trigger event.
2. The PE is coming out of warm reset. Power-on reset is not affected by this erratum.
3. Other specific timing micro-architectural timing conditions occur.

## Implications

If this erratum occurs the PE will deadlock, because it cannot come out of reset correctly. No other PEs in the cluster are affected by this erratum.

## Workaround

Due to the precise timing conditions required to hit this erratum, no general workaround is necessary.

However, if this issue is observed it can be worked around by setting the CTICONTROL.GBLEN bit to 0 to disable triggers.

# 1980125
# A change of ASID without context synchronization might cause memory ordering issues

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A change of Address Space IDentifier (ASID) without a subsequent context synchronization barrier or event might cause ordering issues on a subsequent `CAS` or `CASP` instruction.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core executes an `MSR` instruction changing TTBRn_ELx.ASID, TCR_ELx.A1, or TCR_ELx.AS.
2. The core executes a `CAS` or `CASP` instruction using the translation affected by the ASID change within 6 instructions of the `MSR` above, without an intervening context synchronization barrier or event.

## Implications

If the conditions are met, the `CAS` or `CASP` instruction might violate memory ordering rules and uniprocessor semantics.
The conditions imply changes of ASID, or other translation-related registers, which are not expected to occur for actively used translations. A context synchronization event is expected before resuming execution in a context which has had its translation control registers modified.

## Workaround

An `ISB` instruction between the `MSR` and the `CAS`/`CASP` instructions using the modified translation prevents the erratum from occurring.

## 1980599
## An ECC error in the L1 data cache might not be detected

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

A store of less than 4 bytes of data to Normal Inner Write-Back, Outer Write-Back Cacheable memory, or a Tag Checked store, might not detect an ECC error when reading the L1 data cache data or Memory Tagging Extension (MTE) RAMs. This might result in data corruption or an incorrect value of the TFSR_ELx register.

### Configurations Affected

This erratum affects configurations with core cache protection enabled.

### Conditions

Set 1:

1. The core executes a store of less than 4 bytes of data to Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. A (potentially speculative) memory access receives a NDErr or DErr response.
3. An error or poisoned data is present in the L1 data RAM.

Set 2:

1. MTE checking is enabled, and SCTLR_ELx.TCF = 0b10.
2. The core executes a Tag Checked store to Normal Inner Write-Back, Outer Write-Back Cacheable memory.
3. A (potentially speculative) memory access receives a NDErr or DErr response.
4. An error or poisoned data is present in the L1 data MTE RAM.

### Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the conditions in set 1 are met, data corruption may occur at the 4-byte-aligned 4-byte quantity around the location written to by the store.

If the conditions in set 2 are met, TFSR_ELx might be updated with an incorrect value.

## Workaround

There is no workaround.

# 1981723
# A continuous stream of DVM operations from another PE might block core powerdown

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A continuous stream of Distributed Virtual Memory (DVM) operations from a PE not within the complex, might block the powerdown of a core in a complex.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The system is attempting to power down a core in the complex.
2. A continuous stream of DVM operations is sent from a PE not within the complex.

## Implications

If the above conditions are met, the core might not be able to power down, and the P-Channel handshake might be stalled until the stream of DVM operations stops.

## Workaround

There is no workaround.

# 1982956
# SIMD and floating point loads to non-gatherable device memory might over-read

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

SIMD and floating point loads to non-gatherable device memory might over-read by reading a 32-byte-aligned 32-byte quantity around the accessed bytes.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following condition:

1. The core executes a SIMD or FP load to non-gatherable device memory, with all accessed bytes within a 16-byte-aligned 16-byte granule.

## Implications

If the condition is met, the memory access generated for the load will over-read by reading a 32-byte-aligned 32-byte quantity rather than staying within the permitted 16-byte-aligned 16-byte quantity.

## Workaround

Software might be able to avoid the use of SIMP or FP loads to device memory within 32-bytes of read-sensitive memory.

## 1986930
## Corruption might occur on MTE allocation tags

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

MTE allocation tags might be corrupted on certain systems.

### Configurations Affected

This erratum affects all configurations. This erratum affects systems with **BROADCASTOUTER**=1 that do not have a snoop filter, or have an imprecise snoop filter.

### Conditions

1. The complex has a cache line cached in a shared state without MTE allocation tags.
2. A core executes a Tag Unchecked store to the cache line, causing a line upgrade coherency request (MakeReadUnique).
3. The line is not lost at the requestor, but the MakeReadUnique request returns a response with data and allocation tags.
4. Unlikely, but not rare, timing-sensitive microarchitectural conditions occur.

### Implications

If the above conditions are met, the stored allocation tags for the given cache line might be corrupted. When a coherent requestor in the same shareability domain later requests allocation tags, it might observe the data corruption.

The last condition requires the interconnect to return data and MTE tags in response to a MakeReadUnique request with TagOp=Invalid, even though the requestor did not lose the cache line. This is not expected to happen in systems with a precise snoop filter. The return of MTE allocation tags in response to a TagOp=Invalid MakeReadUnique request likely requires the presence of a system cache, tag cache, or multiple clusters in the system.

This erratum is categorized as Category C because no systems using this revision of the core meet the above properties.

### Workaround

There is no workaround.

# 1987125
# ERR2MISC0.OFR and ERR2MISC0.OFO might not be set on counter overflow

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Configurations Affected

This erratum affects configurations with core cache protection enabled.

## Conditions

This erratum occurs under the following condition:

1. A detected and corrected error causes ERR2MISC0.CECO or ERR2MISC0.CECR to overflow.

## Implications

There is still substantial benefit to be gained from using the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the condition is met, ERR2MISC0.OFO and ERR2MISC0.OFR might not be set to 1, and it might not be possible to detect an overflow of ERR2MISC0.CECO and/or ERR2MISC0.CECR.

## Workaround

There is no workaround.

# 1987184
# DBG_RECOV or WARM_RST power modes might lead to deadlock or data corruption

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

Due to this erratum, some types of Warm reset may cause cores in the cluster to deadlock or to cause corruption or deadlock of Utility Bus or Advanced Peripheral Bus (APB) transactions. The types of Warm reset affected are those that do not require the cluster to be quiescent when asserted.

## Configurations Affected

All configurations are affected.

## Conditions

1. One of the following power modes is requested in the core Power Policy Unit (PPU):
    - DBG_RECOV with PPU_PTCR.DBGRECOV_PORST_EN set to 0
    - WARM_RST
2. The PE is simultaneously handling at least one of the following:
    - An `MSR` or `MRS` instruction
    - A Utility bus access from outside the cluster
3. Other specific micro-architectural timing conditions occur.

## Implications

Under these conditions, one of more cores in the cluster may deadlock after coming out of Warm reset. The Utility bus or debug APB bus may observe corrupted transactions or deadlock.

## Workaround

The type of Warm reset required to cause this erratum is not expected to be used in normal operation. Instead, it is expected to be used for specific debugging purposes. As such, no workaround is expected to be required.

## 1990749
## Errors or poison in the L2 data RAM might lead to deadlock and/or data corruption

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If both halves of a cache line stored in the L2 cache see an ECC error or have been stored with poison, a subsequent read might lead to a deadlock and/or data corruption.

### Configurations Affected

This erratum affects configurations where the complex is configured with an L2 cache.

### Conditions

The erratum occurs under the following conditions:

1. An L1 instruction or data cache refill hits in the L2 cache.
2. Both 32-byte parts of the cache line stored in the L2 cache read by the refill above see an ECC error or have been stored as poisoned due to an earlier deferred error.

### Implications

If the conditions are met, a deadlock or data corruption on the cache line filled into the L1 cache might occur. Two errors on the same cache line are unlikely to occur. This is not expected to be a significant issue for sample silicon.

### Workaround

There is no workaround.

## 1991004
## Uncontainable error injection via ERR2PFGCTL might occur at the wrong time

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

If uncontainable error injection is enabled via ERR2PFGCTL, these injections might occur at an unexpected time and at a higher rate than expected.

### Configurations Affected

This erratum affects configurations with core cache protection enabled.

### Conditions

1. Injection of uncontainable errors is enabled in ERR2PFGCTL (ERR2PFGCTL.UC = 0b1 and ERR2PFGCTL.CDNEN = 0b1).

### Implications

If the conditions are met, uncontainable error injections might take place at a much greater rate than expected. Errors might be injected whenever the counter reaches zero, without requiring a triggering access. As a result, ERR2MISC0.INDX and ERR2MISC0.WAY might contain invalid information.

### Workaround

Software might be able to avoid injecting uncontainable errors via ERR2PFGCTL.

# 1991342
# Traps of System registers in Debug state might cause unexpected exceptions

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

Secure debug can be disabled by setting EDSCR.SDD. This will trap execution in EL2, EL1, or EL0 in Debug state of any instruction that is configured by EL3 control registers to trap to EL3. The core has the IMPLEMENTATION DEFINED behavior that this trap has priority over other traps to EL2 or EL1. There are certain groups of EL1 System registers that can be configured to trap to both EL2 and EL3. Because of this erratum, while the Processing Element (PE) is at EL1 and in Debug state with Secure debug disabled, traps for these System registers will not behave as expected while they have an EL2 trap enabled.

If a trap to EL3 is also enabled, the Secure debug disabled undefined exception has priority. Undefined exceptions at EL1 can target EL1 or EL2, depending on whether HCR_EL2.TGE is set. However, this erratum causes the exception to always target EL2 in this case, regardless of HCR_EL2.TGE.

If the trap to EL3 is disabled, and the EL2 trap is the next highest priority exception, the syndrome will indicate an undefined exception instead of a System register trap.

The affected System register groups are:

- Interrupt controller EL1 registers (ICC_*, ICV_*).
- LORegion registers.
- RAS error record registers.
- Pointer Authentication key registers.
- Registers trapped by SCR_EL3.ATA and HCR_EL2.ATA.

## Configurations Affected

All configurations are affected.

## Conditions

The erratum occurs under the following conditions:

1. The PE is in Debug state at EL1.
2. Secure debug is disabled by EDSCR.SDD.
3. One of the EL1 System registers mentioned previously, which can be trapped to EL2 or EL3, is accessed while the EL2 trap is enabled.

4. If the EL3 trap is enabled, HCR_EL2.TGE is not set.

## Implications

The resulting exception under these conditions will either have a different syndrome, or higher target Exception level than expected.

## Workaround

Either enabling Secure debug, or disabling the affected System register traps to EL2 will avoid any cases of unexpected exception syndrome or target Exception level.

# 1996476
# A Tag Checked SVE non-fault or first-fault load might hang if an External abort occurs

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A Tag Checked *Scalable Vector Extension* (SVE) non-fault or first-fault load might hang if precise *Memory Tagging Extension* (MTE) checking is enabled and an External abort occurs.

## Configurations Affected

This erratum affects configurations where the **BROADCASTMTE** pin is set to 1.

## Conditions

1. MTE checking is enabled and set to generate synchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b01).
2. A store instruction is executed to Tagged Normal Inner Write-Back, Outer Write-Back Cacheable memory.
3. A Tag Checked SVE non-fault or first-fault load instruction is executed whose active elements overlap with the memory locations written to by the previous store instruction.
4. The load is to Tagged Normal Inner Write-Back, Outer Write-Back Cacheable memory.
5. The load misses in the cache, and a cache refill for the cache line receives a NDErr or DErr response.

## Implications

If these conditions are met, the SVE non-fault or first-fault load might hang.

## Workaround

There is no workaround.

# 1996886
# ELA connected to warm reset instead of cold reset

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

The ELA is connected to warm reset instead of cold reset. This means that any access to the ELA registers will stall while reset is asserted, which can be an issue for the power modes that assert warm reset.

## Configurations Affected

This erratum affects all product configurations with ELA support. Note that this is affecting only the ELA in the complex and not the DSU ELA.

## Conditions

The erratum occurs under the following conditions:

1. Warm reset is asserted.
   - This happens in {OFF_EMU, WARM_RST} power modes or
   - in DBG_RECOV mode if PPU_PTCR.DBG_RECOV_PORST_EN is set to 0 or
   - in {ON, FUNC_RET} if the core sets RMR_EL3.RR bit to 1 requesting a warm reset.
2. There is a register access to ELA registers.

## Implications

If the conditions above are met, the ELA will not be able to trace signals during warm reset, and the related ELA registers will be initialized to their reset values. Additionally, any ELA register access will stall until the core comes out of reset and then it completes normally. However, the ELA read register requests will return the reset register values.

Since the ELA access will not complete until the core leaves warm reset, it could cause a system deadlock.

## Workaround

There is no workaround.

## 1997011
## Asynchronous Tag Check fail not synchronized based on SCTLR_ELx.ITFSB for exceptions in Debug state

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

### Description

When taking an exception in Debug state, Tag Check Faults due to instructions executed before the exception (that are reported asynchronously) are not synchronized into the TFSRE0_EL1 and TFSR_ELx registers based on SCTLR_ELx.ITFSB.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. The core enters Debug state.
2. Tag Check Faults are configured to be reported asynchronously and PSTATE.TCO == 0.
3. A Tag Checked instruction executes which generates a Tag Check Fault.
4. An exception entry occurs to ELx when SCTLR_ELx.ITFSB == 1.

### Implications

If the above conditions are met, then a Tag Check fail due to an instruction executed in step 3 of the conditions will not be synchronized into the TFSRE0_EL1 and TFSR_ELx registers upon taking the exception in step 4.

### Workaround

A debugger should execute a `DSB` before reading TFSRE0_EL1 or TFSR_ELx.

# 1998835
# Double-bit errors in the duplicate L1 data cache tag RAMs might lead to loss of coherency or deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

If a double-bit error is detected on a read of the duplicate L1 data cache tag RAMs, a deadlock might occur.

## Configurations Affected

This erratum affects configurations with core cache protection enabled.

## Conditions

The erratum can occur under two sets of conditions.

Set 1:

1. A `DC ISW`, `DC CSW`, or `DC CISW` operation to the L1 data cache is executed. No error is detected on the read of the duplicate L1 data cache tag RAMs.
2. The same core performs an L1 data cache refill for a line in the same set as the set/way operation above. This refill might be speculative.
3. A double-bit error occurs on a read of the duplicate L1 data cache tag RAMs for the L1 cache refill above.

Set 2:

1. A new or deferred error is detected in the L1 data cache tag, data, dirty, or Memory Tagging Extension (MTE) RAMs. This causes a hardware set/way operation being generated.
2. For the hardware-initiated set/way maintenance operation above, no error is detected on the read of the duplicate L1 data cache tag RAMs.
3. The same core performs an L1 data cache refill for a line in the same set as the set/way operation above. This refill might be speculative.
4. A double-bit error occurs on a read of the duplicate L1 data cache tag RAMs for the L1 data cache refill above.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. If the conditions are met, coherency might be lost or a deadlock might occur on later memory accesses. Double-bit errors on the duplicate L1 data cache tag RAMs are classed as uncontainable, and Arm believes these implications are in line with those of an uncontainable error. The detection and reporting of the error are unaffected.

## Workaround

The first set of conditions can be avoided by software. The use of data cache maintenance by set/way operations to the L1 data cache is not necessary as the core performs automatic cache maintenance on powerup and powerdown. Where software intends to use set/way operations regardless, the data cache should be turned off to ensure the intended behavior of cleaning the cache and avoiding a speculative refill to the cache during the sequence.

The second set of conditions does not have a workaround. However, a double-bit error occurring in the duplicate tag RAMs on a cache line that has previously had another error in the L1 data cache is very unlikely.

# 1999032
# Non-Data Errors on memory accesses not attributable to a core might not be reported

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

If a Non-Data Error occurs on a memory request not attributable to a core, such as a cache eviction writeback, an L2-allocating read, or an L2 prefetch, the error might not be reported.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The complex performs a cache eviction writeback, an L2-allocating read, an L2 prefetch, or other access not attributable to a core.
2. The memory access above receives a NDErr response.

## Implications

If these conditions are met, the NDErr response might not be reported in the ERR2STATUS error record register. The downstream component which generated the NDErr response should have reported the error.

## Workaround

Arm does not believe a workaround is needed, as the error should have been reported by the downstream memory component.

# 2000000
# Interrupt might not be taken in finite time

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A pending, unmasked interrupt may not be taken in finite time.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. An interrupt is pending.
2. The interrupt is unmasked according to the table in section "Asynchronous exception masking" in the Arm Architecture Reference Manual.
3. A continuous stream of instructions is executed, consisting exclusively of either of the following two instructions:
   - `MSR TRCPRGCTLR`
   - Branch instructions
4. Micro-architectural timing conditions occur.

## Implications

If the above conditions are met, then the interrupt pending in step 1 of the conditions might not be taken while the instructions in step 3 are being continuously executed.

## Workaround

There is no workaround.

# 2006188
# L2 cache debug operations might hang when no L2 cache is present

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

If an L2 cache debug operation is performed when no L2 cache is present, the complex might hang.

## Configurations Affected

This erratum affects configurations without an L2 cache (parameter L2_CACHE is set to FALSE).

## Conditions

1. A core performs a cache debug operation to the L2 cache (`SYS IMP_CDBGL2CDR` / `SYS IMP_CDBGL2CMR`).

## Implications

The cache debug operation might hang. Subsequent System register accesses from either core in the complex might hang.

## Workaround

Software may be able to avoid L2 cache debug operations when no L2 cache is present. The specific details are system-dependent.

# 2012183
# Double-bit errors in the duplicate L1 data cache tag RAMs might lead to deadlock on a snoop

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

Double-bit errors in the duplicate L1 data cache tag RAMs might lead to a deadlock on a snoop.

## Configurations Affected

This erratum affects configurations with core cache protection enabled (CORE_CACHE_PROTECTION set to TRUE).

## Conditions

The erratum occurs under the following conditions:

1. A TLB hardware update of the dirty bit or access flag or an L2-allocating write from the CPU misses in the complex and starts a read. No error is detected on the read of the duplicate L1 data cache tag RAMs.
2. The L2 receives a snoop for the same cache line.
3. A double-bit error occurs on a read of the duplicate L1 data cache tag RAMs for the L1 cache refill above.
4. The DSU or interconnect downstream of the complex prevents completion of the first read until the snoop completes.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in the overall system failure rate due to this erratum. If the conditions are met, the snoop might not progress, causing a deadlock. Arm believes that this is rare as the location in the L1 data cache tag RAMs must not have had any error on the first read, but has a double-bit error on a read only shortly after.

## Workaround

There is no workaround.

# 2015240
# Clearing a pending OS Unlock Catch debug event might cause unpredictable behavior

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

When an OS Unlock Catch debug event is pending, writing a value of 0b1 to EDESR.OSUC under complex and rare microarchitectural timing conditions might cause unpredictable behavior to be observed.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. OS Unlock Catch debug events are enabled (CTIDEVCTLR.OSUCE == 1).
2. The state of the OS lock changes from locked to unlocked.
3. A debugger writes 0b1 to EDESR.OSUC.
4. Exact microarchitectural timing conditions occur.

## Implications

If these conditions are met, then unpredictable behavior in the core might occur, including execution of incorrect instructions or at an incorrect Exception level. This includes, but is not limited to, incorrect entry to EL0 and/or incorrect switching to non-secure state.

PSTATE.IL will always be set when these conditions are met.

Security state escalation is not possible when these conditions are met.

## Workaround

There is no workaround.

# 2016064
# A continuous stream of memory requests from one CPU in the complex might block another

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

## Description

A continuous stream of memory requests from one core in the complex might block an L1 cache refill of another.

## Configurations Affected

This erratum affects configurations with two cores in the complex.

## Conditions

1. One of the cores in the complex executes a continuous stream of memory requests. This can be any combination of stores, TLB maintenance instructions, I-cache maintenance instructions, **`PRFM`** instructions, as well as TRBE-generated trace data writes.
2. The other core in the complex is attempting an L1 cache refill, either due to a demand or speculative request.
3. A majority of memory requests from the first core experience unusually long delays.

## Implications

If the above conditions are met, the L1 cache refill might not progress until the stream of operations from the first core is interrupted.

## Workaround

A denial of service on a secure process can be avoided if the secure OS sets up a timer-based interrupt source to interrupt all cores periodically.

## 2017921
## Exception or DCPS3 instruction in debug state might block execution of instruction from EDITR

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

### Description

If an exception or `DCPS3` instruction is executed in debug state then an instruction written to the External Debug Instruction Transfer Register (EDITR) might never be executed.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. The core enters Debug state.
2. A debugger writes a first instruction to the EDITR and one of the following occurs:
    - The first instruction is a `DCPS3` instruction that switches to EL3 from a lower Exception level.
    - The first instruction generates an exception in Debug state, that targets ELx with SCTLR_ELx.ITFSB == 1.
3. A debugger writes a second instruction to the EDITR.
4. Exact microarchitectural timing conditions occur.

### Implications

If these conditions are met, then the instruction written to the EDITR in step 3 will not be executed. Subsequent writes to the EDITR will be blocked due to EDESR.ITE == 0

If these conditions are met then exit from debug state will still be possible.

It is unlikely that this could be observed in a real system, due to debugger latency between step 2 and step 3 of the conditions.

### Workaround

There is no workaround.

## 2025809
## A double-bit error on the L1 data cache MTE tag RAMs might result in a missed or incorrect error record in ERR1STATUS

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

### Description

If a double-bit error is detected on the L1 data cache Memory Tagging Extension (MTE) tag RAM, no error might be recorded in ERR1STATUS, or it might be recorded incorrectly.

### Configurations Affected

This erratum affects configurations with core cache protection enabled (CORE_CACHE_PROTECTION set to TRUE).

### Conditions

The erratum might occur under two sets of conditions.

Set 1:

1. MTE checking is enabled and set to generate asynchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b10).
2. A store instruction is executed to Inner Write-Back and Outer Write-Back, tagged, memory.
3. A double-bit error, or poison because of a previously deferred error, is detected on the L1 data cache MTE tag RAM.
4. The double-bit error is still present on a subsequent correction read of the L1 data cache MTE tag RAM.

Set 2:

1. MTE checking is enabled and set to generate asynchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b10).
2. A store instruction is executed to Inner Write-Back and Outer Write-Back, tagged, memory.
3. A double-bit error, or poison because of a previously deferred error, is detected on the L1 data cache MTE tag RAM.
4. The double-bit error is no longer present on a subsequent correction read of the L1 data cache MTE tag RAM.

### Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the conditions in set 1 are met, the error might be reported as a deferred error, but the TFSR_ELx register might not have been updated correctly due to the store instruction.
If the conditions in set 2 are met, no error might be reported, but the TFSR_ELx register might not have been updated correctly due to the store instruction.

As a result, both might result in a false negative in TFSR_ELx.

## Workaround

There is no workaround.

# 2033473
# A hardware update of the dirty bit might occur speculatively

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

A hardware update of the dirty bit might be speculatively performed for a store or atomic that sees a permission fault because of Privileged Access Never (PAN), or an alignment or stack alignment fault.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under two sets of conditions:

Set 1:

1. The core is executing in EL1, or EL2 with Virtualization Host Extensions (VHE) enabled (HCR_EL2.E2H = 0b1).
2. PSTATE.PAN is set.
3. An **`STTR`** instruction is executed speculatively, but never committed.
4. A committed privileged store or atomic instruction to the same address is executed and sees a permission fault because of PAN.
5. Hardware update of the dirty bit is enabled for the page of memory accessed by the store or atomic instruction above.

Set 2:

1. A store or atomic instruction is executed speculatively, but never committed, or a load instruction is executed.
2. A committed privileged store or atomic instruction to the same page is executed, and sees an alignment fault or stack alignment fault.
3. Hardware update of the dirty bit is enabled for the page of memory accessed by the store or atomic instruction above.

## Implications

If the conditions are met, the access permissions of the descriptor might be speculatively updated to include write permissions (dirtied), even though the store sees a permission fault because of PAN, or an alignment fault, or a stack alignment fault. For most systems, including those anticipated to be used by sample silicon, having the dirty bit speculatively in the presence of a permission failure does not lead to any incorrect operation.

## Workaround

There is no workaround.

# 2035302
# CONTEXTIDR_EL2 breakpoint matching is enabled at EL3

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

Architecture issue D17428 describes that breakpoints performing context comparisons against CONTEXTIDR_EL2 should not match when executing at EL3. The core implements the originally documented behavior and will match at EL3 if EL2 is enabled in the current Security state.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is executing at EL3.
2. Halting debug is allowed.
3. EDSCR.HDE == 1.
4. OSLSR_EL1.OSLK == 0.
5. SCR_EL3.NS == 1 or SCR_EL3.EEL2 == 1.
6. DBGBCR<n>_EL3.BT is configured to one of the following:
    - Unlinked CONTEXTIDR_EL2 match (DBGBCR<n>_EL3.BT == 0b1100).
    - Unlinked Full Context ID match (DBGBCR<n>_EL3.BT == 0b1110).
7. DBGBVR<n>_EL3 and CONTEXTIDR_EL2 contain the same value.

## Implications

If the above conditions are met, then a context breakpoint might generate a debug event based on CONTEXTIDR_EL2 while executing at EL3.

## Workaround

To avoid this erratum, ensure that any breakpoints performing context comparisons against CONTEXTIDR_EL2 are configured not to match at EL3, by ensuring that either DBGBCR<n>_EL1.HMC is 0 or DBGBCR<n>_EL1.SSC[0] is 1.

# 2036369
MPAM3_EL3.TRAPLOWER reset on Cold reset

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

## Description

Architecture issue D17395 documents a discrepancy in the specified reset domain for the MPAM3_EL3.TRAPLOWER bit. The core implements the incorrect interpretation, resetting MPAM3_EL3.TRAPLOWER on Cold reset rather than Warm reset.

## Configurations affected

All configurations are affected.

## Conditions

1. MPAM3_EL3.TRAPLOWER is set to 0.
2. The core is Warm reset.
3. An access to one of the MPAM System registers is made from EL2 or lower.

## Implications

If these conditions are met, the MPAM3_EL3.TRAPLOWER bit will not be set to 1 on the Warm reset, and so accesses to MPAM registers from EL2 and lower Exception levels will not be trapped to EL3.

## Workaround

Boot software should initialize the MPAM3_EL3.TRAPLOWER bit and not rely on the reset value.

# 2048342
# A load or store instruction might hang when encountering multiple uncorrectable or deferred errors

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

A load or store instruction to Normal Cacheable memory might hang if the instruction accesses more than one cache line and at least one cache line being accessed is poisoned due to an earlier deferred error.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The core executes a load instruction or a Tag Checked store instruction, other than a load or store of vector registers.
2. The load or store crosses a 64-byte boundary, accessing two cache lines.
3. Both cache lines accessed by the instruction encounter uncorrectable or deferred errors.

## Implications

If the conditions are met, the instruction executed in step 1 might hang, preventing further execution until the error is cleared. Interrupts will not be taken.

## Workaround

There is no workaround.

# 2052205
# ERR2STATUS.SERR might be incorrect after an NDErr response is received

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

The value of ERR2STATUS.SERR might be corrupted if an NDErr response is received. The ERR2STATUS.UE field is not affected.

## Configurations Affected

Only configurations with core cache protection disabled (CPU_CACHE_PROTECTION = 0) are affected.

## Conditions

The erratum occurs whenever a non-attributable NDErr response is received by a complex.

## Implications

If the conditions are met, the ERR2STATUS register is updated correctly, except for the ERR2STATUS.SERR field. The ERR2STATUS.UE bit will indicate that an uncorrected error occurred.

## Workaround

There is no workaround.

# 2052374
# PMU event ST_RETIRED might be inaccurate for Store-Exclusive instructions

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

The processor implements the performance monitor event ST_RETIRED to count executed memory-write instructions. Due to this erratum, a Store-Exclusive instruction might cause the ST_RETIRED event to count in cases where it normally should not.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. One of the PMU event counters is configured to count event 0x0007, ST_RETIRED.
2. Either:
    - A Store-Exclusive instruction is executed which generates a Data Abort, or
    - A Store-Exclusive instruction is speculatively executed in the shadow of an instruction generating a Data Abort.

## Implications

Any count of the ST_RETIRED event might be higher than expected. This is expected to have an insignificant effect on the use of the ST_RETIRED event for performance analysis.

## Workaround

There is no workaround.

# 2053387 Instructions in Debug state after a watchpoint debug event might not be executed

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

When the core enters Debug state, instructions written to the External Debug Instruction Transfer Register (EDITR) might never be executed.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Halting debug is enabled (EDSCR.HDE == 1).
2. Halting is allowed.
3. The OS Lock is unlocked (OSLSR_EL1.OSLK == 0).
4. An enabled watchpoint is hit, causing the core to enter Debug state.
5. A debugger writes an instruction to the EDITR.
6. Exact microarchitectural timing conditions occur.

## Implications

If these conditions are met, then the instruction written to the EDITR in step 5 will not be executed. Subsequent writes to the EDITR will be blocked due to EDSCR.ITE == 0.

If these conditions are met, then exit from Debug state will still be possible.

## Workaround

If a workaround is required, if the instruction written in step 5 has not been executed in a macroscopic amount of time, the debugger should leave Debug state and reenter via the same watchpoint.

# 2054370
# Execution of instructions in Debug state after a watchpoint debug event might incorrectly set EDSCR.ERR and corrupt PSTATE

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

Execution of a specific set of instructions while in Debug state due to a watchpoint debug event might lead to EDSCR.ERR being set, and to PSTATE.PAN and PSTATE.UAO being corrupted.

## Configurations Affected

All configurations are affected.

## Conditions

1. Halting debug is enabled (EDSCR.HDE == 1).
2. Halting is allowed.
3. The OS Lock is unlocked (OSLSR_EL1.OSLK == 0).
4. An enabled watchpoint is hit, causing the core to enter Debug state.
5. While in Debug state, a debugger writes an instruction to the EDITR. The instruction is one of the following types:
    - "Memory access instructions at various access sizes" as defined by section "Executing instructions in Debug state" in the Arm Architecture Reference Manual
    - Load/Store Register (PAC) - All variants
    - Load/Store Memory Tags - All variants
    - System Instructions - All variants
    - Data Synchronization Barrier
    - Data Memory Barrier
    - Error Synchronization Barrier
    - `MSR` or `MRS` to one of the following System registers:
        - LORSA_EL1
        - LOREA_EL1
        - LORN_EL1
        - LORC_EL1
        - MAIR_ELx
        - TCR_ELx
        - TRBBASER_EL1
        - TRBLIMITR_EL1
        - TRBMAR_EL1

- TRBPTR_EL1
- TRBSR_EL1
- TRBTRG_EL1
- TTBR0_ELx
- TTBR1_ELx
- VSTCR_EL2
- VSTTBR_EL2
- VTCR_EL2
- VTTBR_EL2
- GIC registers (ICC_* or ICH_*)
- ETE registers (TRC*)
- Implementation defined registers (IMP_*, accessed via S3_* encodings)

6. The instruction in step 5 is executed without generating a synchronous exception.

There is no limit on the number of instructions executed in Debug state between steps 4 and 5.

No synchronous exceptions were generated between steps 4 and 5.

No `DCPS<n>` or `DRPS` instructions were executed between steps 4 and 5.

## Implications

If these conditions are met, then the value of PSTATE.PAN and PSTATE.UAO might be corrupted. EDSCR.ERR will also be set to 1.

## Workaround

If a watchpoint debug event causes entry to Debug state, then before writing any other instructions to EDITR, the debugger should execute a `DCPS<n>` instruction, first saving ELR_ELx, SPSR_ELx, ESR_ELx, DLR_EL0 and DSPSR_EL0 as necessary.

# 2056307 A Tag Checked store exclusive might hang if an external abort occurs

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

A Tag Checked Store-Exclusive instruction might hang if precise Memory Tagging Extension (MTE) checking is enabled and an external abort occurs.

## Configurations Affected

This erratum affects configurations where the **BROADCASTMTE** pin is set to 1.

## Conditions

The erratum occurs under the following conditions:

1. A store or atomic is executed to a memory location A. The memory location is mapped as Tagged Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. MTE checking is enabled and set to generate synchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b01).
3. A Load-Exclusive instruction is executed to the same memory location A.
4. An L1 data cache refill for the cache line containing memory location A receives a NDErr or DErr response.
5. A Tag Checked Store-Exclusive to memory location A is executed.
6. Timing-sensitive micro-architectural conditions occur.

## Implications

Substantial benefit is still being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. If the conditions are met, the Store-Exclusive might hang.

## Workaround

There is no workaround.

# 2059297
# PSTATE.TCO might be unchanged for debug entry due to a watchpoint debug event

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

Entry to debug state due to a watchpoint debug event will not set PSTATE.TCO.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Halting debug is enabled (EDSCR.HDE == 1).
2. Halting is allowed.
3. The OS Lock is unlocked (OSLSR_EL1.OSLK == 0).
4. An enabled watchpoint is hit, causing the core to enter Debug state.

## Implications

The architecture requires that PSTATE.TCO is set on entering Debug state. If these conditions are met, then the value of PSTATE.TCO will be unchanged when entering debug state and therefore might not be set.

## Workaround

If a watchpoint debug event causes entry to debug state, the first instruction pushed by the debugger should be **`MSR TCO, #1`**. Alternatively, a **`DCPS<n>`** instruction will set PSTATE.TCO.

# 2071968
# Incorrect context ID might be associated with trace data

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

An **`MSR`** to either CONTEXTIDR_EL1 or CONTEXTIDR_EL2 might lead to the incorrect context identification information being associated with a trace packet.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. The core enters Debug state.
2. ETM tracing is enabled (TRCPRGCTLR.EN == 1).
3. An **`MSR`** instruction is executed which updates the active CONTEXTIDR_ELx register.
4. The core exits Debug state.

The ordering of steps 2 and 3 has no impact.

There are no P0 instructions executed between steps 3 and 4.

OR:

1. The core is reset.
2. ETM tracing is disabled (TRCPRGCTLR.EN == 0).
3. The core enters Debug state.
4. ETM tracing is enabled (TRCPRGCTLR.EN == 1).
5. The core exits Debug state.

There are no p0 instructions executed between steps 4 and 5.

There are no **`MSR`** instructions executed which update the active CONTEXTIDR_ELx register.

OR:

1. ETM tracing is disabled (TRCPRGCTLR.EN == 0).

2. Either of the following:
   - An **MSR** instruction is executed which updates the active CONTEXTIDR_ELx register.
   - No instructions have been traced since the last core reset.
3. ETM tracing is enabled (TRCPRGCTLR.EN == 1).
4. The core executes an **ESB**.
5. A virtual or physical SError is pending and is unmasked according to the table in section "Asynchronous exception masking" in the Armv8-A Architecture Reference Manual.
6. Exact microarchitectural timing conditions occur.

There are no P0 instructions executed between steps 2 and 4.

## Implications

If these conditions are met, then a trace element might have the incorrect context identification information associated with it, potentially leading to the generation of incorrect (or missing) context elements by the trace unit.

## Workaround

For the first set of conditions, an **ISB** should follow step 3.

No workaround is required for the second set of conditions.

There is no workaround for the third set of conditions.

# 2077160
# Incorrect ordering after change in cacheability

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

## Description

If the memory type of an address region is changed from Cacheable to Non-cacheable, and then back again, and rare micro-architectural conditions occur, then stale data might be observed in a cache.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. An address region of memory is marked in the translation tables as Write-Back Cacheable memory.
2. The hardware prefetcher starts a data prefetch to an address within this region.
3. The translation tables are updated to change the memory type to Non-cacheable or Device memory. This would involve a break-before-make sequence.
4. A sequence of cache clean and invalidate instructions are executed to ensure that any Cacheable data in the memory region does not remain in the caches.
5. Rare, timing-sensitive micro-architectural conditions occur, causing the cache clean and invalidate to not clean and invalidate the line brought in by the hardware prefetch.
6. A core or other component in the system writes to the region that is now marked Non-cacheable or Device.
7. The translation tables are changed a second time to mark the memory as Write-Back Cacheable again.
8. A load instruction is executed. The load might observe the stale data that was prefetched into the cache, rather than the Non-cacheable data that was written.

## Implications

The above sequence is very specific and would typically take a very long time to execute. It requires that the hardware prefetch is started before the translation table modification, and then does not complete until after both the translation table modification and the cache maintenance have finished.

In addition, rare micro-architectural conditions must occur for the hardware prefetch to not be affected by the cache maintenance. Therefore, the combination of these conditions is going to be extremely rare. Furthermore, the change in memory type implies a change of use of the memory, and many such changes of use will not require preservation of the data between uses.

## Workaround

This erratum has no workaround.

# 2077274
# Extra A-sync packet might get written to Trace Buffer in Trace prohibited region

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

An external trace analyzer can request that an A-sync packet is injected into the trace stream, which the Embedded Trace Extension (ETE) will insert when the next P0 Element is traced. Due to this erratum, this A-sync packet might incorrectly get generated and written to trace buffer memory via TRBE under the conditions mentioned below.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. TRBE is enabled.
2. ETE is in a trace prohibited region.
3. A `TSB` instruction is executed and completed.
4. TRBE is disabled.
5. A synchronization request is received on the ATB interface.
6. TRBE is enabled.

## Implications

If the above conditions occur, an A-sync packet might go to TRBE and when a new `TSB` instruction is executed, this packet might get written to memory. Under normal usage Arm expects that this unexpected trace will have no impact.

## Workaround

There is no workaround.

# 2085871
# Shareability aliases might result in incorrect Tag Check Faults

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

A page mapped as Normal, Tagged memory under both a shareable and a non-shareable mapping might result in a spurious or missed Tag Check Fault.

## Configurations Affected

This erratum affects all configurations where the **BROADCASTMTE** pin is HIGH.

## Conditions

The erratum occurs under the following conditions:

1. A page of physical memory is mapped under both a shareable and a non-shareable mapping. Both mappings are for Normal memory, Tagged, Inner-Writeback and Outer-Writeback cacheable.
2. MTE checking is enabled (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn != 0b00).
3. The core has a location A, cached using the non-shareable mapping above.
4. Another PE in the system uses the shareable mapping to update the tags in memory for a location A, and the update becomes globally observable.
5. The core executes a data cache maintenance operation with the intent of invalidating the non-shareable cached entry for location A.
6. The core executes a Tag-checked store or atomic to location A, without a `DMB` between the previous CMO and this store.

## Implications

If the above conditions are met, the Tag-checked store is not guaranteed to see the update to the tags at location A, and might use the previous tags for its tag check. This might result in a spurious or missed Tag Check Fault.

## Workaround

Software might be able to avoid the use of cacheable non-shareable, or aliased shareability, for tagged memory. Alternatively, software can insert a `DMB LD` after the cache maintenance operation to ensure ordering.

# 2088637
# A watchpointed SVE load might update the FFR register incorrectly

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

An SVE load might update the FFR register incorrectly after a watchpoint hit.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. Watchpoints are enabled and a watchpoint is set on a memory location A.
2. The core speculatively executes an SVE load or store to memory location A in the shadow of a mis-predicted branch. This instruction is not architecturally executed.
3. The core executes an unaligned SVE Non-Fault or First-Fault load crossing a 32-byte boundary, to location A, or another location with a watchpoint. The watchpointed location is in the upper 32-byte aligned quantity.

## Implications

If the conditions are met, the fields of the FFR register corresponding to the watchpointed elements of the load in condition 3 might be not be cleared. This will effectively mean that the watchpoint is ignored for that load. Data written into the architectural register file is unaffected.

## Workaround

This erratum has no workaround.

# 2092740
# Debug state exit after execution of a DRPS instruction might lead to deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

A debug restart request following the execution of a `DRPS` instruction might lead to deadlock.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The core enters debug state.
2. The external debugger passes an `DRPS` instruction to the core via the EDITR.
3. The core executes the `DRPS` instruction which enters Illegal Execution State via one of the following:
    - An illegal return event
    - A legal return event that sets PSTATE.IL to 1
4. The external debugger triggers a restart request trigger event.

## Implications

If the conditions are met, then the PE will deadlock.

## Workaround

A debugger should observe that any `DRPS` instruction issued through the EDITR was completed and did not cause entry to Illegal Execution State, before issuing a restart request.

# 2096738
# Double-bit errors in the duplicate L1 data cache tag RAMs might lead to deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

A double-bit error in the duplicate L1 data cache tag RAMs might lead to a deadlock if a double-bit error is detected towards the end of a hardware cache flush sequence.

## Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION enabled.

## Conditions

1. A core in the complex is being powered down, and a hardware cache flush sequence is started.
2. Towards the end of the hardware cache flush sequence, a double-bit error is detected on a read of the duplicate L1 data cache tag RAMs for an access other than the hardware cache flush, for example caused by an access from the other core in the complex.
3. The double-bit error was not present/detected when the hardware cache flush request accessed the same RAM location shortly before.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. If the above conditions are met, the complex might deadlock, and a hardware watchdog might be required as the error cannot be handled by software. Double-bit errors on the duplicate L1 data cache tag RAMs are classed as uncontainable, and Arm does not believe this erratum materially degrades the benefit from ECC.

## Workaround

There is no workaround.

## 2112025
## Instructions might be missed for tracing purposes before Debug state entry due to a watchpoint hit

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

### Description

When a halting watchpoint hit causes the core to enter Debug state, [E1:E0] field of the exception packet for Debug entry might be incorrect and programmed address comparators might not match.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. Halting debug is enabled (EDSCR.HDE == 1).
2. Halting is allowed.
3. The OS Lock is unlocked (OSLSR_EL1.OSLK == 0).
4. ETM tracing is enabled (TRCPRGCTLR.EN == 1). This can be enabled either by software or an external debugger.
5. Tracing in the current region is unprohibited.
6. A p0 instruction is executed and traced by the trace unit.
7. Some execution occurs at the target of the p0 instruction in step 6.
8. An enabled watchpoint is hit, causing the core to enter Debug state.
9. Exact microarchitectural timing conditions occur.

### Implications

If these conditions are met, [E1:E0] field of the exception packet for Debug entry might be incorrect and comparators programmed on instructions at step 7 might not match.

### Workaround

There is no workaround for the incorrect [E1:E0] field.

Comparator issues are not observable if they are programmed to match only branch instructions.

# 2120833
# DISR_EL1 access in Debug state might be incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

An **`MSR`** or **`MRS`** instruction to DISR_EL1 while in Debug state might write or return the incorrect value.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. The core enters Debug state.
2. The core is executing in EL1 or EL2.
3. SCR_EL3.EA == 1
4. One of the following occurs:
    a. DISR_EL1 is written, using the **`MSR`** instruction.
    b. DISR_EL1 is read, using the **`MRS`** instruction.

## Implications

If step 4.1 of these conditions is met, then DISR_EL1 will not be updated.

If step 4.2 of these conditions is met, then the read of DISR_EL1 will read zero.

## Workaround

If a workaround is required and EL3 is accessible by the debugger, then a debugger should change the PE state to EL3, clear SCR_EL3.EA, and then return to the Exception level in step 2 before accessing DISR_EL1.

# 2133769 PMU event 0x000C PC_WRITE_RETIRED might be inaccurate for ERET instructions

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

The processor implements the performance monitor event PC_WRITE_RETIRED to count software changes of the Program Counter (PC). Due to this erratum, the PC_WRITE_RETIRED does not count for `ERET` instructions.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. One of the PMU event counters is configured to count event `0x000C`, PC_WRITE_RETIRED.
2. An `ERET` instruction is executed.

## Implications

Any count of the PC_WRITE_RETIRED event might be lower than expected. This is expected to have an insignificant effect on the use of the PC_WRITE_RETIRED event for performance analysis.

## Workaround

There is no workaround.

# 2135690
# PMU events based on STALL_FRONTEND might be inaccurate

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

The processor implements several performance monitor events to count attributable frontend stalls caused by other events. Due to this erratum, the attributable frontend stall events might not be counted.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Any PMU event counter is configured to count at least one of the attributable STALL_FRONTEND_* events
   0x00e1, STALL_FRONTEND_CACHE
   0x00e2, STALL_FRONTEND_TLB
   0x00e3, STALL_FRONTEND_PDERR
2. No PMU event counters are configured to count 0x0020, STALL_FRONTEND
3. The ETM trace unit is not enabled

## Implications

Any count of the attributable STALL_FRONTEND_* events might be lower than it should be.

## Workaround

The attributable STALL_FRONTEND_* events will count as expected if counting is also enabled for 0x0020, STALL_FRONTEND, or by enabling the ETM trace unit.

# 2141037
# OFF_EMU to OFF power mode transition might result in a deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

If a core in a two-core complex enters the OFF_EMU power mode, then it might become unable to move to OFF power mode. The second core in the complex would then become unable to power down.

## Configurations Affected

This erratum affects all configurations with two cores in a complex

## Conditions

The erratum occurs under the following conditions:

1. One core in the complex is in the OFF_EMU power mode.
2. The second core in the complex is in ON/FUNCRET/FULLRET power modes.
3. ETM is enabled by setting TRCPRGCTLR.EN.
4. One of the following occurs:
    - The first core is requested to move to OFF power mode.
    - The second core is requested to move to OFF_EMU/OFF power modes or is reset by the RMR.RR sequence.

## Implications

1. The core in OFF_EMU power mode might become unable to enter OFF power mode.
2. The other core in the complex would then be unable to do RMR.RR reset or enter OFF_EMU/OFF power modes. The requests would be denied repeatedly and could lead to a deadlock.

## Workaround

ETM can be disabled on entering into OFF_EMU state by setting 1'b0 in TRCPRGCTLR.EN. In this case Trace debugging cannot be used in OFF_EMU state.

## 2146058
## PMU events counts might be inaccurate

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r1p0.

### Description

The processor implements several performance monitor events to count attributable front-end stalls caused by other events. Due to this erratum, multiple Performance Monitoring Unit (PMU) events might not be counted.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs under any of the following conditions:

1. Any PMU event counter is configured to count at least one of the attributable STALL_FRONTEND_* events (0x00e1, STALL_FRONTEND_CACHE; 0x00e2, STALL_FRONTEND_TLB; 0x00e3, STALL_FRONTEND_PDERR) and:
   - No PMU event counters are configured to count 0x0020, STALL_FRONTEND.
   - The ETM trace unit is not enabled.
2. One of the PMU event counters is configured to count event 0x000C, PC_WRITE_RETIRED, and an `ERET` instruction is executed.
3. One of the PMU event counters is configured to count event 0x00ED STALL_BACKEND_VPU_HAZARD

### Implications

Any count of the attributable STALL_FRONTEND_*, PC_WRITE_RETIRED, events might be lower than it should be. Any count of the attributable STALL_BACKEND_VPU_HAZARD event might be lower or higher than it should be.

### Workaround

The attributable STALL_FRONTEND_* events will count as expected if counting is also enabled for 0x0020, STALL_FRONTEND, or by enabling the ETM trace unit. There is no workaround for the other events.

# 2161448
# PMU_HOVFS event not always exported when self-hosted trace disabled

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2 and r0p3. Fixed in r1p0.

## Description

PMU_HOVFS is a Performance Monitoring Unit (PMU) event that can be exported to the Embedded Trace Macrocell (ETM). Exporting this event is disabled if TRFCR_EL2.E2TRE is set to 0b1, but this setting only applies when self-hosted trace is enabled. Due to this erratum, the event is never exported when TRFCR_EL2.E2TRE is set to 0b0, including when self-hosted trace is disabled.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The ETM is configured to use PMU_HOVFS as an external input event.
2. Self-hosted trace is disabled and TRFCR_EL2.E2TRE is set to 0b0.

## Implications

Overflows of PMU counters reserved by Exception level 2 (EL2) might not be visible.

## Workaround

To use the PMU_HOVFS as an external input event when self-hosted trace is disabled, ensure TRFCR_EL2.E2TRE is set to 0b1.

# 2164605 A watchpoint hit while disabling Halting debug might cause an incorrect debug exception to be taken

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

If a watchpoint debug event occurs at the same time as halting debug is disabled, a debug exception might be taken, even though debug exceptions are disabled.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. Debug must be enabled (DBGEN == 1).
2. Halting debug is enabled (EDSCR.HDE == 1).
3. Debug exceptions are disabled (MDSCR_EL1.MDE == 0).
4. Halting is allowed.
5. The OS lock becomes set by changing OSLAR_EL1.OSLK from 0 to 1.
6. An enabled watchpoint is hit.
7. Exact micro-architectural timing conditions occur.

No context synchronization occurs between OS lock being set and the watchpoint being hit.

## Implications

If these conditions are met, then a watchpoint exception might be taken, even though this is forbidden because MDSCR_EL1.MDE == 0.

## Workaround

This erratum can be avoided by implementing the following:

1. After an `MSR` to OSLAR_EL1, an `ISB` should be executed before any memory instructions which might generate a watchpoint.

2. Before an APB write to OSLAR_EL1, all watchpoints should be disabled by clearing DBGWCR<n>_EL1.E for each watchpoint.

## 2175229
## A single ECC error on the L2 data RAMs might be double-counted

### Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

### Description

ERR2MISC contains counter information on errors reported by L2. Counter information might be unreliable under some conditions.

### Configurations Affected

All configurations with CORE_CACHE_PROTECTION set to True are affected.

### Conditions

This erratum occurs under the following conditions:

1. An ECC error is detected in the L2 DATA RAM.
2. Heavy memory traffic occurs in the L2 cache.
3. Complex micro-architectural conditions occur.

### Implications

If these conditions occur, an error for the L2 DATA RAM might be reported more than once, incorrectly increasing the value of ERR2MISC0 counters.

### Workaround

This erratum has no workaround.

# 2181318
# A PE trace unit trigger might not be reflected in TRBSR_EL1 after a TSB

## Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

## Description

If a TRBE (Trace Buffer Extension) trigger event because of TRBTRG_EL1 occurs during a TSB in a prohibited region, the register state in TRBSR_EL1 after the TSB might not reflect the trigger event.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. TRBE is enabled and running.
2. TRBLIMITR_EL1.TM is not set to 0b11 (ignore trigger).
3. The core executes a TSB in a prohibited region.
4. The PE trace unit detects a trace trigger.

## Implications

If the above conditions are met, then the value of TRBSR_EL1.TRG, TRBSR_EL1.S and TRBSR_EL1.IRQ might not reflect that a trigger event occurred. Arm expects it to be an uncommon occurrence that a trace trigger is detected during the TSB in a prohibited region.

## Workaround

There is no workaround.

# 2187223
# In Halting debug state, the core might take an illegal execution exception when it should not

## Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

## Description

The PSTATE.IL bit is set when the Core executes an illegal exception return. Attempting to execute any instructions when this bit is set will cause the Core to take an exception.
Due to this erratum, the PSTATE.IL bit is not cleared when entering the Halt state in certain circumstances, causing an incorrect exception to be generated.

## Configurations Affected

This erratum affects all configurations that include support for the AArch32 Instruction Set when at the EL0 exception level.

## Conditions

This erratum occurs under the following conditions:

1. The core in Halt state at EL0
2. DSPSR register is modified to trigger an illegal exception return
3. The core exits Halt state, triggers an illegal exception return, and sets PSTATE.IL
4. The core enters Halt state due to a CTI trigger
5. The core executes an instruction while in Halt state via the Instruction Transfer Register

There must be no instructions executed between step 3 and 4.

## Implications

If these conditions are met, then an illegal execution state exception will be taken when it should not. Even if an illegal execution state exception is taken when it should not, the value of PSTATE.IL is restored properly when the core exits the Halt state after step 5.

## Workaround

There is no workaround.

# 2189260
# An MMU fault might not be correctly reflected in the Trace Buffer Extension TRBSR_EL1 register

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

If the Trace Buffer Extension is enabled and an MMU fault occurs on the last page of the trace buffer, the MMU fault might not be correctly recorded in TRBSR_EL1.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. TRBE is enabled and running.
2. The trace buffer is set to fill mode (`TRBLIMITR_EL1.FM = 0b00`).
3. The last page of the trace buffer region is invalidated, and modified in a way that would result in an MMU fault. For example, this could be marking the descriptor as invalid or clearing the access flag or removing write permissions.
4. Simultaneously, the trace buffer limit is reached.

## Implications

If the conditions are met, the pointer value in TRBPTR_EL1 will reflect the correct address at which the MMU fault occurred, but TRBSR_EL1.EC and TRBSR_EL1.MSS might not reflect an MMU fault. Instead, TRBSR_EL1.EC and TRBSR_MSS might reflect that the trace buffer filled.

Arm expects that commonly, an operating system would pin the pages used for the trace buffer, avoiding any MMU faults from occurring in the first place.

## Workaround

No workaround is expected to be required, based on common usage models of the trace buffer which involves pinning the pages.

# 2189707 Clean MTE tag writeback might occur after a store to the cache line

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

A store to a cache line might incorrectly dirty the MTE tags, and result in a later writeback of both data and MTE tags.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs if all the following conditions apply:

1. Tagged memory accesses are enabled (SCTLR_ELx.ATAn = 1).
2. The core executes a store.
3. Uncommon micro-architectural conditions occur.

## Implications

If the conditions are met, the MTE tags on the cache line can be marked as dirty even though they were not written to, resulting in additional unnecessary DRAM writebacks of tagged memory. The micro-architectural conditions are uncommon, so the unnecessary DRAM writebacks are expected to be negligible for performance or power.

Marking the MTE tags on the cache line as dirty without an explicit write prevents the use of tag storage locations for data, as software coherency cannot be maintained.

## Workaround

This erratum has no workaround.

# 2212805
# An ECC error while in the active-not-pending state might clear PSTATE.SS

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

An ECC error (single bit, double bit, or poison) when executing a Load/Store instruction while software stepping in the active-not-pending state might lead to the core entering the active-pending state.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs if all the following conditions apply:

1. An exception return is executed, which enables software stepping (entering the active-not-pending state).
2. Load/Store that generates an ECC error.
3. Microarchitectural timing conditions occur.

## Implications

If the ECC is due to a single bit error, the instruction being stepped will not be executed.
If the ECC is due to a double bit error or poison data, this might result in a persistent error when software stepping.

## Workaround

This erratum has no workaround.

# 2217109
# RAS error reporting might be incorrect on simultaneous errors

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

ERR2STATUS and ERR2MISC registers contain information about reported errors. In case of simultaneous errors reported, the value of those registers might be incorrect.

## Configurations Affected

All configurations with CORE_CACHE_PROTECTION set to True are affected.

## Conditions

Set 1:

- A parity error on the TLB RAMs is detected on the same cycle as a corrected error on the L2DB RAMs.

Set 2:

- Multiple errors are detected in different L2 DATA banks.

## Implications

If the first set of conditions occurs, the reported error might have a corrupted value of SERR, IERR, index, and way in the ERR2STATUS and ERR2MISC registers.
If the second set of conditions occurs, ERR2STATUS.CECO might not be increased correctly.

## Workaround

There is no workaround.

# 2220074
# ETM will indicate that a T32 CC-failed ISB has caused a Context Synchronization Event when it has not

## Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

## Description

When the ETM traces an `ISB` instruction, it should generate an E Atom if the ISB performed a Context Synchronization Event and a N Atom otherwise.

On Cortex-A510, a CC-failed T32 `ISB` will not generate a Context Synchronization Event, but the ETM will generate an E atom anyway.

## Configurations Affected

This erratum affects all configurations that include AArch32 at EL0.

## Conditions

This erratum occurs under the following conditions:

1. The ETM is enabled and tracing is active.
2. The core executes a T32 CC-failed ISB instruction.

The `ISB` can only CC-fail if it's inside an IT block.

## Implications

When reconstructing the trace, it will appear as though a Context Synchronization Event has occurred when it has not.

## Workaround

This erratum has no workaround.

## 2226937
## IMP_CDBGDR0_EL3 read data might be incorrect

### Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

### Description

IMP_CDBGDR0_EL3 is used to perform direct accesses to internal memory. Software first writes to one of a number of system registers to trigger a read of a given cache location, and that value is written into the IMP_CDBGDR0_EL3 register. Due to this erratum, the value written might be incorrect.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs under the following conditions:

1. One of the following operations is executed:
    - SYS IMP_CDBGL2CDR
    - SYS IMP_CDBGL2CMR
    - SYS IMP_CDBGL2CTR
    - SYS IMP_CDBGL2TR0
    - SYS IMP_CDBGL2TR1
    - SYS IMP_CDBGL2TR2
2. The core reads the IMP_CDBGDR0_EL3 register.

### Implications

Due to this erratum, bits [63:48] of the result of each memory read operation come from the previous memory read operation, not the current one. That, is, after executing one of the SYS IMP_CDBGL2* instructions listed above:

Bits [47:0] of IMP_CDBGDR0_EL3 will be correct.

Bits [63:48] of IMP_CDBGDR0_EL3 might be incorrect.

### Workaround

Software should repeat the operation that caused IMP_CDBGDR0_EL3 to be updated before reading it. Bits [47:0] will come from the second operation, and bits[63:48] will come from the first operation. As long as the memory being accessed stays the same between both operations, IMP_CDBGDR0_EL3 will hold the correct value.

# 2230537
# Load following SVE predicated load/store might cause ordering violation

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

Load following Scalable Vector Extension (SVE) predicated load/store might cause ordering violation if there is an older alias store, or I-cache invalidation instruction, or TLB invalidation instruction, and SVE predicated load/store matches programmed watchpoint Virtual Address (VA).

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. Watchpoint is enabled and is programmed with the VA X
2. Store to VA Y and physical address A, or I-cache invalidation instruction, or TLB invalidation instruction
3. SVE predicated load/store matches against a programmed watchpoint with VA X
4. Younger Load with the VA X and physical address A
5. A concurrent store to memory location A executed on another PE. The store is coherence-after the concurrent store of condition 2.
6. Complex micro-architectural conditions.

## Implications

If the conditions are met, the last load might forward data to dependent instructions that is inconsistent with the data written to the architectural register file. This load might forward the old data to dependent instructions, while writing the data of the last store to the architectural register file.

## Workaround

There is no workaround.

# 2236402
# Asynchronous exception or debug event might be taken before an exception catch debug event that was generated by an exception entry

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

An asynchronous exception or debug event might be taken before an exception catch debug event that was generated on exception entry.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is configured to generate an exception catch upon exception entry to ELx.
2. Halting debug is allowed.
3. An exception entry to ELx occurs, which generates an exception catch debug event.
4. One of the following occurs:
    - An interrupt is pending and is unmasked according to the table in the "Asynchronous exception masking" section of the Armv8-A Architecture Reference Manual.
    - An asynchronous halting debug event is pending.
5. Micro-architectural timing conditions occur.

## Implications

If the above conditions are met, then the asynchronous exception or debug event pending in condition 4 might be taken prior to the exception catch debug event in condition 3.

## Workaround

There is no workaround.

# 2241478
# The L1D_CACHE_REFILL and L1D_CACHE_LMISS_RD PMU events might over-count

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

The L1D_CACHE_REFILL and L1D_CACHE_LMISS_RD PMU events might over-count.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following sets of conditions:

Set 1:

1. A load and a store are executed to the same cache line, A, in close proximity to one another.
2. Cache line A is not present in the L1 data cache, and a new cache refill is started.

Set 2:

1. A cache refill for cache line A is ongoing.
2. A load to cache line A misses in the cache but does not start a new refill, but instead reuses the refill above.

## Implications

If the first set of conditions is met, the L1D_CACHE_REFILL event might count twice instead of once.
If the second set of conditions is met, the L1D_CACHE_LMISS_RD event might count loads that miss in the cache but do not start a new refill, and instead are satisfied by an ongoing refill.

## Workaround

There is no workaround.

# 2265919
# Top byte of DLR_EL0 might be incorrect when entering AArch64 halting Debug state

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

In AArch64 state, when related System registers enable Top Byte Ignore (TBI), the top byte of the 64-bit PC value copied to DLR_EL0 when entering halting Debug state should always be `0x00` or `0xFF`. Due to this erratum, the top byte of DLR_EL0 could be updated to other values when entering AArch64 halting Debug state.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs if all the following conditions apply:

1. In AArch64 state, TBI is active, and the top byte of PC is changed due to TBI.
2. AArch64 halting Debug state is entered.

## Implications

A debugger software in AArch64 halting Debug state relying on the value of top byte of DLR_EL0 could execute incorrectly. However, when leaving halting Debug state, execution will restart from the correct address.

## Workaround

When DLR_EL0 is needed by software, the debugger could work out the expected effect of TBI from related System registers TCR_EL1, TCR_EL2, HCR_EL2, and the current exception level. If TBI is active, replace the top byte of DLR_EL0 with 0x00 or 0xFF as calculated from the related System register fields.

## 2266023
## Vector stores might not use a faster forwarding path

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

### Description

In some cases, vector store instructions can be incorrectly prevented from using a faster forwarding path, delaying execution by an additional cycle.
This affects all store instructions that access a single vector register or store more than 128 bits of data, and only occurs in cases where the vector store is not consuming a result from the vector forwarding network, and occurs infrequently in most benchmark code sequences.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. A previously issued vector data processing instruction obtains at least one source operand from the VPU forwarding network
2. A subsequent vector store instruction is allocated to the same issue slot as the previous vector data processing instruction, and is accessing the register file directly for its operands.

### Implications

The vector store instruction requires an additional cycle to execute.

### Workaround

This erratum has no workaround.

# 2271084
# DTR flags not cleared on external debugger access while leaving Debug state

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

The Data Transfer Registers (DTRs) provide a mechanism to transfer data between an external debugger and the core. They consist of write-only registers to transmit data (DBGDTRTX_EL0 and DBGDTRTXint), read-only registers to receive data (DBGDTRRX_EL0 and DBGDTRRXint), and associated data control flags.

Due to this erratum, if these registers are accessed by the external debugger while the debug exit procedure is in progress, then the accesses will go ahead but the flow control flags will not be updated correctly.

## Configurations Affected

This erratum affects all configurations .

## Conditions

1. The debugger requests a debug exit.
2. The debugger does an external access to the DBGDTRRX/ DBGDTRTX, while the debug exit is ongoing.
3. Certain microarchitectural timing conditions are met.

## Implications

For an external read to DBGDTRTX, the EDSCR.TXU and EDSCR.TXfull flags will not be updated.
For an external write to DBGDTRRX, the write will be ignored and the EDSCR.RXO and EDSCR.RXfull flags will not be updated.

## Workaround

There is no workaround.

# 2272274
# Core might execute two instructions on Halting Step debug event

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

Halting Step is a debug resource that a debugger can use to make the core step through code one instruction at a time. Due to this erratum, the core might step through two instructions.

## Configurations Affected

All configurations are affected.

## Conditions

This erratum occurs under the following conditions

1. The debugger must execute a Halting Step debug event.
2. One of the following must occur:
    a. The instruction after the one to be stepped reads or writes a system register, and traps.
    b. The instruction after the one to be stepped reads or writes a system register and generates a software access debug event.
    c. The instruction to be stepped is a WFI or WFE that traps.
    d. The instruction to be stepped is a WFI or WFE that causes the core to sleep.

## Implications

If the instruction after the one to be stepped traps, the core will leave debug state, correctly execute the first instruction, execute the next instruction (taking the trap normally), then enter debug state.

If the instruction after the one to be stepped generates a software access debug event, EDSCR.STATUS will indicate the debug entry was due to a software access to debug register. DLR_EL0 will hold the restart address for the instruction in step 2.2.

If the instruction to be stepped is a WFI or WFE that traps, the core will leave debug state. Correctly execute the WFI/WFE (taking the trap), execute the next instruction, then enter debug state.

If the instruction to be stepped is a WFI or WFE that causes the core to sleep, then the core will leave debug state, execute the WFI/WFE, enter sleep state, leave sleep state as normal, execute the next instruction, then enter debug state.

## Workaround

This erratum has no workaround.

# 2287488
# Core might not execute instruction on Halting Step debug event

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

Halting Step is a debug resource that a debugger can use to make the Core step through code one instruction at a time. The Core will start in Debug state, exit it, execute one instruction, then return to Debug state. Due to this erratum, the Core might re-enter Debug state without executing an instruction.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The debugger must execute a Halting Step debug event.
2. Certain complex microarchitectural conditions occur.

## Implications

The Core will not execute the instruction, but the PC will continue to point to that instruction. On the next Halting step event, the Core will execute that instruction again, and the microarchitectural conditions that caused it to be skipped will no longer apply, causing the instruction to execute normally.

## Workaround

There is no workaround.

# 2289541
# PMU counter overflow can cause spurious PMU_OVFS and PMU_HOVFS events

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

PMU_OVFS and PMU_HOVFS are Performance Monitoring Unit (PMU) events that can be exported to the Embedded Trace Extension (ETE). One of these events is asserted each time a PMU event counter overflows. Which one of the two events that is asserted is determined by whether the counter is reserved for use by EL2, as defined by MDCR_EL2.HPMN. Due to this erratum, when at least one counter is reserved for EL2, a single overflow can assert both events simultaneously. In this case, one of the events is spurious and should not have been asserted for a counter in this range.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The PMU is configured to split the counters into two ranges by setting MDCR_EL2.HPMN to less than PMCR_EL0.N
2. The ETM is configured to use PMU_HOVFS or PMU_OVFS as an external input event.

## Implications

ETM trace can be triggered unexpectedly by overflows in event counters outside of the expected ranges for PMU_OVFS or PMU_HOVFS, if only one of these events is configured to be a trigger.

## Workaround

There is no workaround.

# 2289878
# Core might step two instructions at once

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

Software Step and Halting step are both ways of forcing the processor to execute one instruction at a time.

Due to this erratum, the Core might execute two instructions when it should only execute one.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The Core must be performing Software step or Halting step.
2. The instruction to be stepped must generate a precise exception.
3. Certain microarchitectural conditions occur.

## Implications

The Core will execute two instructions before either taking the Software step exception, or entering Debug state.

## Workaround

There is no workaround.

# 2291784
# PSTATE.IL might be cleared on ERET to Software Stepping

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r1p0. Fixed in r1p1.

## Description

An ERET which enters the Illegal Execution state and enables Software Stepping might lead the core to clear PSTATE.IL.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. An exception return is executed, which enters the Illegal Execution state (PSTATE.IL == 1) and enables Software Stepping (entering the active-not-pending state).
2. Complex microarchitectural timing conditions occur.

## Implications

If these conditions are met, then PSTATE.IL will be cleared following the exception return in step 1. This might cause the core to step an instruction in the Illegal Execution state.

SPSR_ELx.IL for the subsequent Software Step exception will be 0.

## Workaround

This erratum has no workaround.

## 2293834
## The core might report that a load-exclusive instruction has not been stepped when it should

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

### Description

On taking a Software Step exception, the ESR_ELx.EX bit indicates whether a load-exclusive instruction was stepped. Due to this erratum, this bit might not be set when it should.

On entering Debug state due to Halting step, EDSCR.STATUS will be `0b011111` if a load-exclusive instruction was stepped, or `0b011011` for any other instruction. Due to this erratum, this field might be `0b011011` when it should be `0b011111`.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs under the following conditions:

1. The core is performing a Software Step or Halting Step of a load-exclusive instruction.
2. The load-exclusive instruction takes an exception due to a data side abort.
3. One of the following is true:
    - The (Data Abort) exception entry occurs to EL1 when SCTLR_EL1.IESB == 1.
    - The (Data Abort) exception entry occurs to EL1 when SCTLR_EL1.ITFSB == 1.
4. MDCR_EL2.TDE is 1 and either the core is executing in Non-secure state or Secure EL2 is enabled.

### Implications

THe ESR_ELx.EX or EDSCR.STATUS fields might be incorrect.

### Workaround

This erratum has no workaround.

# 2300878
# Software stepping ESB might set SPSR.ELx.SS to incorrect value

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

An SError when software stepping an ESB in the active-not-pending state, fails to set SPSR_ELx.SS.

## Configurations Affected

All configurations are affected.

## Conditions

This erratum occurs under the following conditions:

1. An exception return is executed, which enables software stepping (entering the active-not-pending state).
2. An `ESB` is executed.
3. A virtual or physical SError is pending and is unmasked according to the table in the "Asynchronous exception masking" section of the Armv8-A Architecture Reference Manual.
4. Micro-architectural timing conditions occur.

## Implications

If these conditions are met, then the value of SPSR_ELx.SS after taking the SError exception will be 0 rather than 1.

## Workaround

This erratum has no workaround.

# 2303522
# Synchronous tag checking on a store exclusive might cause a deadlock if double bit/fatal error seen in L1-Duplicate RAM

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

A double-bit ECC error in the Duplicate L1 data cache tag RAM might cause a deadlock on an exclusive sequence if synchronous tag checking is enabled.

## Configurations Affected

This erratum affects all configurations where the **BROADCASTMTE** pin is HIGH.

## Conditions

The erratum occurs under the following conditions:

1. MTE checking is enabled and set to generate synchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b01).
2. The core executes an exclusive sequence, consisting of a tag-checked load exclusive and a tag-checked store exclusive.
3. A double bit ECC error is detected in the Duplicate L1 data cache tag RAM on the same index/way as the cache line that the exclusive sequence is executing on, and this error is detected between the load exclusive and store exclusive execution.
4. Unlikely, timing-sensitive micro-architectural conditions occur.

## Implications

If the conditions are met, the store exclusive might deadlock the core. There is still substantial benefit to be gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

## Workaround

This erratum has no workaround.

# 2316094
# MTE checking might not be reliable

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

MTE checking might not be reliable in certain configurations.

## Configurations Affected

This erratum affects configurations where the complex is configured without an L2 cache (the configuration parameter L2_CACHE is FALSE) and the BROADCASTMTE pin is HIGH.

## Conditions

This erratum occurs under the following conditions:

1. MTE checking is enabled and set to generate asynchronous exceptions (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn = 0b10).
2. A store instruction is executed to Inner Writeback and Outer Writeback, tagged, memory.
3. Another overlapping store is executed with a different tag for the overlapping bytes of memory, shortly after the first store.

## Implications

If the conditions are met, the tag mismatch might not be detected, resulting in an incorrect value of TFSR_ELx.

Arm does not expect that the resulting minor increase in false-negative tag checks have a noticeable impact on the system.

## Workaround

This erratum has no workaround.

# 2321712
# DLR_EL0[1] is ignored when performing debug exit to A32

## Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0 and r1p1. Fixed in r1p2.

## Description

Upon leaving the Debug state, the new PC based off DLR_EL0. PC[0] is set to 0b0, PC[31:2] are given by DLR_EL0[31:2], and PC[1] depends on DLR_EL0[1] and the DSPSR.

When the DSPSR indicates a target of A32, PC[1] should be given by DLR_EL0[1], but due to this erratum, it is always set to 0b0.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is in Debug state.
2. The DSPSR indicates a return state of AArch32 (DSPSR_EL0.M[4] == 0b1).
3. The DSPSR indicates a return ISA of A32 (DSPSR_EL0.T =0b0).
4. A write to DLR_EL0 sets bit[1].
5. The core leaves Debug state and starts executing A32 instructions.

## Implications

PC[1] will be incorrectly set to 1'b0, and the Prefetch Abort will be not generated as expected.

## Workaround

There is no workaround.

## 2324165
## Processor state might be corrupted if EDSCR.INTdis is set while asynchronous interrupt is in flight

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

### Description

The EDSCR.INTdis bit disables interrupts when not in debug state. If an external debugger sets this bit in between an asynchronous interrupt being signaled to the core and that interrupt being taken, then the state of the processor might be corrupted.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs under the following conditions:

1. Invasive debug is enabled for the current security state.
2. ExternalInvasiveDebugEnabled() is TRUE.
3. One of the the following asynchronous interrupts is indicated to the core, but has not yet been taken:
    - FIQ
    - IRQ
    - SError
    - Virtual FIQ
    - Virtual IRQ
    - Virtual SError
4. An external debugger write sets EDSCR.INTdis.
5. Certain micro-architectural timing conditions occur.

### Implications

The exception will be taken incorrectly and might have the following effects:

- Exception level might be incorrect (The exception might be taken to EL0).
- PC might be incorrect.
- CPSR.M might be incorrect.
- ESR_EL1, ESR_EL2 or ESR_EL2 might be incorrect.

- FAR_EL1, FAR_EL2 or FAR_EL2 might be incorrect.

## Workaround

The external debugger should only set EDSCR.INTdis when the processor is halted.

## 2331844
## Software stepping ERET might set PSTATE or SPSR to incorrect value

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

### Description

When software stepping an ERET, an SError due to an IESB might corrupt certain PSTATE or SPSR fields.

### Configurations Affected

All configurations are affected.

### Conditions

This erratum occurs under the following conditions:

1. An exception return is executed which enables software stepping (entering the active-not-pending state).
2. An ERET is executed.
3. SCTLR_ELx.IESB is set.
4. A virtual or physical SError is pending and is unmasked according to the table in the "Asynchronous exception masking" section of the Armv8-A Architecture Reference Manual.
5. Micro-architectural timing conditions occur.

### Implications

If these conditions are met, then after taking the SError exception, the value of PSTATE or SPSR might be incorrect.

The following SPSR fields can be affected: TCO, PAN, D, A, I, and F.

The following PSTATE field can be affected: PAN.

This is unlikely to cause issues for most software as the SError will be treated as fatal by the handling software, so the value of SPSR_ELx and PSTATE will be unused.

### Workaround

This erratum has no workaround.

# 2335678
# BFMMLA/VMMLA result might be corrupted when forwarding source operand from vector load

## Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0 and r1p1. Fixed in r1p2.

## Description

When a BFMMLA or VMMLA is receiving forwarded data from an older vector load, the result of the BFMMLA/VMMLA might be incorrect.

## Configurations Affected

This erratum affects any configuration using the 2x128-bit VPU datapath. Configurations using the 2x64-bit datapath are unaffected.

## Conditions

1. The program contains a sequence of at least 9 BFMMLA/VMMLA instructions with no direct data interdependencies.
2. A vector load is present within the BFMMLA/VMMLA sequence that writes to the Zd/Vd register of at least 1 earlier BFMMLA/VMMLA instruction.
3. Either a second vector load is present within the BFMMLA/VMMLA sequence that writes to the Zd/Vd register of at least 1 later BFMMLA/VMMLA instruction, or a BFMMLA/VMMLA instruction is present reading from (and writing to) the same Zd/Vd register as the earlier BFMMLA/VMMLA and vector load.
4. No vector store or other vector data processing instructions are present in the sequence.
5. Precise microarchitectural timing conditions occur.

## Implications

If these conditions are met, the result of a BFMMLA/VMMLA in the instruction sequence might be incorrect.

## Workaround

No workaround is required as this instruction sequence will not occur in real code.

# 2341012
# Physical SError interrupts while software stepping a load or store instruction might cause two instructions to be stepped

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

When software stepping a load or store instruction, the core might execute two instructions when it should only execute one.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. An exception return is executed, which enables software stepping (entering the active-not-pending state).
2. A load or store instruction is executed.
3. A physical SError is pending and is unmasked according to the table in the "Asynchronous exception masking" section of the Armv8-A Architecture Reference Manual.
4. An exception is taken due to the physical SError, which is taken to an Exception level that debug exceptions are enabled from.
5. Certain microarchitectural conditions occur.

## Implications

The core will execute two instructions before taking the Software Step exception.

## Workaround

This erratum has no workaround.

## 2342004
## TLB Parity Check cannot be disabled

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

### Description

A system using the cores cannot disable parity check on TLB RAMs.

### Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION set to 1.

### Conditions

ERR2CTRL.ED is set to 0b0, disabling error detection in L2 and TLB RAMs.

### Implications

If the previous conditions are met, errors in TLB RAMs will still be detected, affecting the values of ERR2STATUS and ERR2MISC registers.
Interrupts generated by parity errors in TLB RAMs can be masked clearing ERR2CTLR.CFI.

### Workaround

This erratum has no workaround.

## 2342918
## PrefetchTgt transactions are generated when PrefetchTgt is disabled

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

### Description

PrefetchTgt transactions are generated even when PrefetchTgt is disabled for instruction fetch memory access, through control bits in IMP_CMPXECTLR_EL1[37:36].
The default setting of these control bits is to not generate PrefetchTgt transactions.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. A memory access instruction for instruction fetch to normal memory, other than Inner-Writeback and Outer-Writeback cacheable, that results in an L1 and L2 miss.
2. PrefetchTgt transactions for instruction fetch memory accesses are disabled with IMP_CMPXECTLR_EL1[37:36] = 0b00.

### Implications

If the conditions are met, a PrefetchTgt transaction might be generated even though the control bits disable their use.
Support of PrefetchTgT is a requirement for the core, and unexpected generation of PrefetchTgt transactions should not cause functional issues in a system supporting them.
The increased number of PrefetchTgt transaction generated by this erratum is limited to non-cached instruction fetched, which should have a minimal impact on the system.

### Workaround

For the majority of systems, no workaround is needed.
In a system where the use of PrefetchTgt transactions must be disabled for instruction fetch memory access, the software can set IMP_CMPXECTLR_EL1[39:38] = 0b00 (which disables PrefetchTgt generation also for Load-Store accesses); for example by using the following sequence:

```
MRS x0, S3_0_C15_C1_7
MOV x1, #3
BFI x0, x1, #38, #2
```

```
MSR S3_0_C15_C1_3, x0
```

# 2360589
# Exception catch might not be taken on ERET from EL3 to any Non-secure EL

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

Exception catch debug events can be configured to cause entry to Debug state on certain exception entries and exception returns. The level of invasive debug that is permitted is configured by the DBGEN and SPIDEN external inputs. Due to this erratum, if invasive debug is allowed for Non-secure state only, certain exception returns might not cause entry to Debug state when they should.

## Configurations Affected

All configurations are affected.

## Conditions

The erratum occurs if all the following conditions apply:

1. DBGEN = 0b1
2. SPIDEN = 0b0
3. The core is at EL3
4. The core executes an ERET
5. Exception catch on exception return is enabled for the target Exception level
    - For an ERET to ELx, this will be when EDECCR.NSEx == 0b0 and EDECCR.NSRx == 0b1
6. The target of the ERET is Non-secure state

## Implications

If the previous conditions are met, the core will not enter the Debug state when it should.

## Workaround

This erratum has no workaround.

# 2371559
# ESR_ELx might be incorrect after trapped vector instruction in Debug state

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

If Non-secure invasive debug is enabled, but Secure invasive debug is not, then the core is not permitted to enter EL3 when in Debug state. If an exception in debug state would be taken to EL3, the exception will be taken to EL1 or EL2 instead.

Due to this erratum, the exception syndrome reported in ESR_ELx might be incorrect if a vector instruction would be trapped to EL3.

## Configurations Affected

All configurations are affected.

## Conditions

This erratum occurs when the following conditions are true for the sequence described below:

1. DBGEN == 0b1.
2. SPIDEN == 0b0.
3. The core is in Debug state.
4. The core executes either of the following:
   - An SVE instruction that would be trapped to EL3 due to CPTR_EL3.EZ == 0b0, if SPIDEN was 0b1.
   - A SVE, Advanced SIMD or floating point instruction that would be trapped to EL3 due to CPTR_EL3.TFP == 0b1, if SPIDEN was 0b1.

## Implications

Depending on whether the trap due to SPIDEN is taken to EL1 or EL2, ESR_EL1.ISS, or ESR_EL2.ISS will be non-zero when it should be zero.

## Workaround

This erratum has no workaround.

# 2385664
# Core might not execute any instruction when performing Halting Step

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

Halting Step is a debug resource that a debugger can use to make the core step through code one instruction at a time. Due to this erratum, the core might not execute any instruction before returning to debug state.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs when the following conditions are true for the sequence described below:

- DBGEN == 1
- SPIDEN == 0
- Halting Step is enabled by EDECR.SS

Then the following sequence must occur:

1. The core exits Debug State to Non-secure state.
2. The core must execute any of the following instructions that generates an exception targeting EL3:
    - Load
    - Store
    - Pointer Authentication instruction affected by `SCR_EL3.API`
    - A `WFI` instruction affected by `SCR_EL3.TWI`
    - A `WFE` instruction affected by `SCR_EL3.TWE`
    - An `ESB` instruction with a physical SError pending and unmasked according to the table in the "Asynchronous exception masking" section of the Arm Architecture Reference Manual Armv8, for A-profile architecture.
3. The core perform an ERET from EL3 to a Non-secure state.
4. The core starts executing an instruction that will not get an exception targeting EL3.

## Implications

The instruction at step 4 of the above sequence should be executed, then the core should enter debug state. Instead, the core will enter debug state without executing that instruction.

The next time the core attempts to step that instruction, it will be executed, and then the core will enter debug state in the correct manner.

## Workaround

This erratum has no workaround.

# 2393935
# ESR_ELx.EX might be incorrect when stepping a load exclusive

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

When the software is stepping a load exclusive, ESR_ELx.EX might not be set upon taking the corresponding Software Step exception.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. An exception return is executed which enables software stepping (entering the active-not-pending state).
2. A load exclusive is executed.
3. Microarchitectural timing conditions occur.
4. A Software Step exception is taken.

## Implications

If the previous conditions are met, then after taking the Software Step exception, the value of ESR_ELx.EX might be incorrect when ESR_ELx.ISV == 1.

## Workaround

No workaround is needed. As microarchitectural timing conditions are not deterministic, the issue will resolve itself on the loop of the load exclusive.

# 2396657
# ESR_ELx.EX might be incorrect due to asynchronous exception when software stepping

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

When software stepping, an asynchronous exception in the active-not-pending state which causes entry to the active-pending state might set ESR_ELx.EX to the incorrect value.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The core is in Debug state.
2. A load exclusive is executed while in Debug state.
3. Debug exit which enables software stepping (entering the active-not-pending state).
4. A pending asynchronous exception is taken, causing entry into the active-pending state.
5. A Software Step exception is taken.

## Implications

This erratum can only occur if debug exit enables software stepping.
If the previous conditions are met, then after taking the Software Step exception, ESR_ELx.ISV will be set and the value of ESR_ELx.EX might incorrectly indicate that a load exclusive was stepped.

## Workaround

This erratum has no workaround.

# 2423961
# PMU_OVFS event is not generated on PMCCNTR_EL0 overflow

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

The PMU_OVFS event should be generated when any PMU counter overflows. Due to this erratum, this event will not be generated when the cycle counter overflows.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

- The cycle counter is enabled
- The cycle counter is configured to generate an event on an overflow (PMINTENSET_EL1.C == 0b1)

## Implications

ETE activity that is triggered based on this event might not occur.

## Workaround

This erratum has no workaround.

# 2429106
# Exception Catch debug event might not be taken

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

An Exception Catch debug event might not be taken if the exception also generates an Implicit Error Synchronization event (IESB).

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. The core is configured to generate an exception catch upon exception entry to ELx.
2. Halting debug is allowed.
3. An exception entry to ELx occurs, which generates an exception catch debug event.
4. Either of the two following conditions:
   - SCTLR_ELx.IESB == 1
   - SCR_EL3.EA == 1, SCR_EL3.NMEA == 1 and ELx == EL3
5. A physical SError is pending and is unmasked according to the table in the "Asynchronous exception masking" section of the Arm Architecture Reference Manual Armv8, for A-profile architecture.
6. Microarchitectural timing conditions occur.

## Implications

If the previous conditions are met, then the SError might be taken instead of the Debug state entry.

## Workaround

This erratum has no workaround.

# 2429770
# PC_WRITE_SPEC event does not count correctly

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

The PC_WRITE_SPEC event should be generated each time a software change of PC is speculatively executed. Due to this erratum, this event will not count correctly.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following condition:

- One of the PMU counters is configured to count the PMU_WRITE_SPEC event

## Implications

The software will not be able to use the PMU_WRITE_SPEC event for performance analysis.

## Workaround

This erratum has no workaround.

# 2478655
# L2D_CACHE_WB event might over-count

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

L2D_CACHE_WB performance event might be unreliable due to this erratum.

## Configurations Affected

This erratum affects configurations where the DSU is configured as direct-connect (parameter DIRECT_CONNECT is TRUE), and the system is capable of generating protocol-level retries (RetryAck) to a Cortex-A510 CPU.

## Conditions

1. The core receives a protocol-level retry in response to a counted request.

## Implications

L2D_CACHE_WB event might over-count if a request sees a protocol-level retry.

## Workaround

There is no workaround.

# 2601282
# ELADISABLE does not disable APB access to the complex ELA

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

ELADISABLE is a cluster configuration signal. When it is HIGH, it disables access to all ELAs in the cluster. Due to this erratum, if an ELA is configured in the complex, it is still accessible through the debug APB interface when ELADISABLE is HIGH. The ROM table entry for the complex ELA will still be removed, so it is not discoverable by an external debugger.

## Configurations Affected

All configurations with the complex ELA are affected.

## Conditions

The erratum occurs if all the following conditions apply:

1. ELADISABLE is HIGH
2. A Debug APB access is made to the memory-mapped region for the complex ELA

## Implications

This erratum means that the complex ELA cannot be completely disabled. Assuming the correct address offset is already known, a debugger will have free control of the trigger and trace features. The ELA can also stop the core clock, which is its only invasive debug feature. This feature can still be disabled through the authentication interface (DBGEN and SPIDEN), but this also disables all other invasive debug features in the cluster.

## Workaround

This erratum has no workaround.

# 2618004
# LDST_SPEC event might under-count

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

Due to this erratum, the LDST_SPEC performance monitors event does not always increment correctly when a combination of load and store instructions, immediately adjacent to each other in the code, are speculatively executed.

## Configurations Affected

This erratum affects all configurations.

## Conditions

- The PMU is configured to count LDST_SPEC

## Implications

Due to this erratum, the LDST_SPEC performance monitors event might have a lower count than expected.

## Workaround

The sum of the counts for the LD_SPEC and ST_SPEC events can be used as an equivalent to LDST_SPEC event.

# 2636015
# BUS_ACCESS and BUS_ACCESS_WR PMU events are not reliable

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0 and r1p1. Fixed in r1p2.

## Description

BUS_ACCESS and BUS_ACCESS_WR performance events might not be reliable due to this erratum.

## Configurations Affected

This erratum affects all configurations.

## Conditions

No specific conditions are needed.

## Implications

BUS_ACCESS and BUS_ACCESS_WR events are not reliable and should not be used.

## Workaround

This erratum has no workaround.

# 2679270
# External aborts reporting can not be disabled

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, and r1p2. Fixed in r1p3.

## Description

A system able to generate external aborts or poison might see errors reported by the L2 node even if error reporting is disabled.

## Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION enabled (CORE_CACHE_PROTECTION set to 1).

## Conditions

- ERR2CTRL.ED is set to 0b0
- Data or responses are received by the core with Data Error, Non-Data Error, or poison

## Implications

If the previous conditions are met, an error might be reported.
Interrupts generated by the set of conditions can be masked clearing ERR2CTLR.FI and ERR2CTLR.UI.

## Workaround

This erratum has no workaround.

# 2688792
# PMU event L3D_CACHE_LMISS_RD might be inaccurate

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

## Description

Due to this erratum, the performance event 0x400B L3D_CACHE_LMISS_RD might undercount.

## Configurations Affected

This erratum affects all configurations.

## Conditions

No specific conditions are needed.

## Implications

L3D_CACHE_LMISS_RD might undercount.

## Workaround

0x00A2 L3D_CACHE_REFILL_RD can be used instead of L3D_CACHE_LMISS_RD. The implementation of L3D_CACHE_REFILL_RD is equivalent to the definition of L3D_CACHE_LMISS_RD.

## 2690487
## Some architectural PMU events are not always available to trace unit

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

### Description

The PMU architectural events are available to the trace unit through the extended input facility. Due to this erratum, the following architectural events might not be sent to the trace unit after being selected:

- 0x4005 STALL_BACKEND_MEM
- 0x4006 L1I_CACHE_LMISS
- 0x4009 L2D_CACHE_LMISS_RD
- 0x400B L3D_CACHE_LMISS_RD
- 0x4020 LDST_ALIGN_LAT
- 0x4021 LD_ALIGN_LAT
- 0x4022 ST_ALIGN_LAT
- 0x4024 MEM_ACCESS_CHECKED
- 0x4025 MEM_ACCESS_CHECKED_RD
- 0x4026 MEM_ACCESS_CHECKED_WR

In addition, the following architectural events are never sent to the trace unit after being selected:

- 0x8002 SVE_INST_RETIRED
- 0x8006 SVE_INST_SPEC
- 0x8014 FP_HP_SPEC
- 0x8018 FP_SP_SPEC
- 0x801C FP_DP_SPEC
- 0x80E3 ASE_SVE_INT8_SPEC
- 0x80E7 ASE_SVE_INT16_SPEC
- 0x80EB ASE_SVE_INT32_SPEC
- 0x80EF ASE_SVE_INT64_SPEC

### Configurations Affected

All configurations are affected.

### Conditions

- The trace unit is configured to use the extended input facility with an affected event.
- For affected events in the 0x4000-0x403F range, the event is not enabled for counting in the PMU through the PMEVTYPER<n>EL0 registers.

## Implications

The affected events in the 0x4000-0x403F range cannot be used reliably by the trace unit unless the PMU is also configured to count the same event.

The affected events in the 0x8000-0x80FF range cannot be used by the trace unit at all.

## Workaround

This erratum can be avoided for the affected events in the 0x4000-0x403F range if the PMU is configured to count the event selected by the trace unit.

There is no workaround for the affected events in the 0x8000-0x80FF range.

## 2710402
## Read value of IMP_CPUCFR_EL1 might be incorrect

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

### Description

Due to this errata, several fields of the IMP_CPUCFR_EL1 system register always read as 0 instead of their intended value. The affected fields are:

- L2_RAM_EVA
- L2_NUM_RAMCTL_PARTITIONS
- L2_NUM_TAGCTL_SLICES

### Configurations Affected

The values in these fields will match configurations where L2_RAM_EVA is FALSE, and L2_NUM_RAMCTL_PARTITIONS and L2_NUM_TAGCTL_SLICES are both set to 1. All other configurations will have an incorrect value at least one of these fields.

### Conditions

For affected configurations, IMP_CPUCFR_EL1 fields with an incorrect value will always be incorrect.

### Implications

The information provided by this register might not match the configuration of the L2 cache. Software is not expected to rely on these values.

### Workaround

There is no workaround.

# 2713359
# ERR2STATUS.UET field might be incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

## Description

ERR2STATUS register contain information about reported errors. Under some conditions, the value of the UET field might be incorrect.

## Configurations Affected

All configurations are affected.

## Conditions

If CORE_CACHE_PROTECTION = 'b1:

- An external abort is detected on a clean, L2 cache allocating line.
- At the same time, a double-bit error is detected on L2 TAGRAMs or L2 L1 Duplicate TAGRAMs.

If CORE_CACHE_PROTECTION = 'b0:

- An external abort is detected on a clean, L2 cache allocating line.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions occur with CORE_CACHE_PROTECTION = 'b1, the reported error might record UEO instead of UC on ERR2STATUS.UET.
If the condition occur with CORE_CACHE_PROTECTION = 'b0, the reported error might record UC instead of UEO on ERR2STATUS.UET.

## Workaround

There is no workaround required.

# 2718749
# Cache debug target for L2 Data RAM may not record correct data

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

## Description

Cache debug target for L2 data RAM might not record the correct data for the top half of a cache line.

## Configurations Affected

All configurations with CORE_CACHE_PROTECTION set to False are affected.

## Conditions

This erratum occurs if the software executes a cache debug read targeting the L2 data RAM using SYS #6, C15, C4, #3{, <Xt>}.

## Implications

The cache debug data recorded in the IMP_CDBGDR0_EL3 register for the L2 data RAM top half cache line will always be 0, and not reflect the value in the data RAM.

## Workaround

There is no workaround required.

# 2736240
# ERR2STATUS might be incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1 and r1p2. Fixed in r1p3.

## Description

The ERR2STATUS register contains information about reported errors. Under some conditions, the value of the register might be incorrect.

## Configurations Affected

All configurations are affected.

## Conditions

If CORE_CACHE_PROTECTION = FALSE:

- A L2 allocating line is received with poison.

If CORE_CACHE_PROTECTION = TRUE:

- A write to the ERR2STATUS register is performed.

## Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the condition with CORE_CACHE_PROTECTION = FALSE is met:

- The PN field is incorrectly set to 0.

If the condition with CORE_CACHE_PROTECTION = TRUE is met:

- The write is performed while the OF field is set and the write is not clearing it.

## Workaround

There is no workaround required.

# 2834345
# Direct access to internal memory might not be reliable

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, and r1p3. Open.

## Description

Internal memory used by structures in the L2 cache can be read using system registers. In some conditions, the read value might not reflect the value stored in the memory.

## Configurations Affected

This erratum affects configurations having the parameter CORE_CACHE_PROTECTION set to TRUE.

## Conditions

1. A core is active.
2. A stream of cache debug operations is issued, using system registers IMP_CDBGL2CMR and IMP_CDBGL2CDR, while the memory system is processing loads and stores to Normal memory.
3. An *Error Correcting Code* (ECC) error is detected in the L2 DATA RAM.
4. Complex microarchitectural timing conditions occur.

## Implications

Arm expects that the memory system is in a quiescent state while direct access to memory is being performed.
If the conditions occur, the values read while directly accessing internal memory might not be correct.

## Workaround

No workaround is required.

# 2856823
# Speculative dirty bit hardware update might happen for store operation

## Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0, r1p1, r1p2, r1p3. Open.

## Description

Speculative dirty bit hardware update might happen during a store operation.

## Configurations Affected

This erratum affects configurations where the CORE_CACHE_PROTECTION parameter is TRUE.

## Conditions

This erratum occurs under the following conditions:

1. Load operation A
2. Store operation B, following load operation A
3. Hardware update of the dirty bit is enabled for the page of memory accessed by the store operation B
4. Load operation A encounters a potentially correctable ECC error in the L1 data cache, the load operation is microarchitecturally replayed. The erratum occurs If during the replay, the load sees an abort that was not present in the original execution.
5. Timing sensitive microarchitectural condition happens

## Implications

If the previous conditions are met, the Store operation might update the AP[2] or S2AP[1] bit as writable when it should not.

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

## Workaround

No workaround is required for this erratum.

## 2867016
## An uncontainable error might deadlock the cluster

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3. Open

### Description

An uncontainable error detected when a core is doing a line upgrade might deadlock the cluster.

### Configurations Affected

This erratum affects all configurations.

### Conditions

1. A core is doing a store that requires a line upgrade.
2. Unlikely, timing-sensitive, microarchitectural conditions occur, including an uncontainable error detected in the L1 Duplicate Tag RAMs.

### Implications

There is still substantial benefit being gained from the *Error Correcting Code* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions occur, the complex might deadlock.

### Workaround

There is no workaround.

## 2867018
## Error record registers indicate incorrect feature support in configurations without cache protection

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, and r1p3. Open.

### Description

The following error record feature registers contain information about the features implemented in the corresponding RAS node. The affected nodes are Node 1 (L1 memory system) and Node 2 (L2 memory system).

- ERR1FR
- ERR1PFGF
- ERR2FR
- ERR2PFGF

In configurations without cache protection, these registers incorrectly indicate that cache protection is present. This also applies when reading the **ERXFR_EL1** and **ERXPFGF_EL1** registers while **ERRSELR_EL1** is selecting any of the affected nodes. Indication of support for error types caused by External aborts is unaffected.

Also, in the **ERR1PFGCTL** and **ERR2PFGCTL** pseudo-fault generation control registers, fields which control the injection of cache protection error types can be read and written as if support is present. Attempting to inject an unsupported error type will have no effect on pseudo-fault generation.

### Configurations Affected

All configurations with the **CORE_CACHE_PROTECTION** parameter set to FALSE are affected.

### Conditions

The incorrect read and write behavior of these registers in affected configurations is always present.

### Implications

In configurations without cache protection, software might incorrectly assume from these register values that it is actually present. This might lead to an unexpected error injection test result, if a test attempts to inject an error type which it thinks is supported, but no error record is created.

## Workaround

This erratum has no workaround. Contact Arm for more details on which error types are not supported when cache protection is not present.

# 2878694
# LDG or MTE checked load/store might fail to detect poisoned data

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, r1p3. Open.

## Description

An LDG or MTE checked load/store might fail to detect poisoned data.

## Configurations Affected

This erratum affects configurations with BROADCASTMTE=TRUE.

## Conditions

The erratum occurs under the following conditions:

1. An older STG or store operation to cache line X.
2. A younger LDG or MTE checked load/store to the same cache line X.
3. Cache line X has a deferred error.
4. Timing sensitive, microarchitectural conditions occur.

## Implications

If the conditions are met, the LDG or MTE checked load/store might fail to detect poisoned data, and might return an incorrect result for an LDG, or an incorrect tag check result for a checked load or store. If asynchronous MTE tag checks are enabled, the state of TFSR_ELx might get corrupted.

## Workaround

No workaround is required for this erratum.

# 2879976
# Unmodified cache line might be written back to memory

## Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, and r1p3. Open.

## Description

A cache line might be temporarily marked as modified, which might result in that line being written back to memory.

## Configurations affected

This erratum affects all configurations with parameter BROADCASTMTE set to True.

## Conditions

This erratum occurs under the following conditions:

1. Memory location A is marked as Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. The core allocates location A into the L1 data cache or the L2 cache in Unique state without MTE tags. This allocation might be due to committed instructions, speculative execution, or data prefetching.
3. The core executes an STG, ST2G, or STGM that requires fetching of MTE tags by the L2.
4. Another *Request Node* (RN) is requesting the line, and this request is ordered before the previous condition (3) in the *Home Node* (HN). The core will provide the line as dirty, but data remains unchanged.

## Implications

If the previous conditions are met, the cache line for memory location A might be marked as modified, but the data remains unchanged.
If the line is evicted to memory while set as dirty with unchanged data, it will then overwrite the value in memory. If an agent in the system has written to location A through a Non-cacheable mapping, these writes might then be overwritten with the older data from the core cache write-back, causing these writes to no longer be visible. This might then result in data corruption for software-managed coherency use cases.
The scenario is a race conditions where an old value of location A can be temporarily seen by a Non-cacheable observer. If the core executes a read to memory location A before the store, that is requiring a DC IVAC (Data or unified Cache line Invalidate by VA to PoC), the race condition is resolved, and the erratum is not applicable.

## Workaround

No workaround is required for this erratum.

# 2971621
# Unmodified page table cache lines might be written back to memory

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r1p0, r1p1, r1p2, and r1p3. Open.

## Description

Hardware management of the Access Flag or Dirty State might set a cache line containing page table data as dirty even if the descriptor data has not been modified.

## Configurations affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. Hardware management of the Access Flag is enabled (TCR_ELx.HA or VTCR_EL2.HA are set to 0b1) or Hardware management of the Dirty State is enabled (TCR_ELx.HD or VTCR_EL2.HD are set to 0b1).
2. The *Processing Element* (PE) schedules an Access Flag or Dirty State update after having read a descriptor.
3. The descriptor changes before the hardware update is performed, and one of the following conditions applies to the new descriptor:
    - It is not a page or block descriptor anymore.
    - It does not have the DBM bit set.
    - It has different permissions than the old descriptor.

## Implications

If the conditions are met, the descriptor will not be modified, but the line will be marked as dirty. If the stage 1 hardware update of the access flag or dirty bit fails because of a stage 2 fault, the issue does not occur.

The cache line will be written back to main memory when evicted from the PE, which can result in a negligible increase in system power.

## Workaround

No workaround is required.

## 2976328
## Store data might be lost when a correctable error is detected in the L1 data cache

### Status

Fault Type: Programmer Category C
Fault Status: Present in r1p1, r1p2, and r1p3. Open.

### Description

Store data might be lost when a correctable error is detected in the L1 data cache RAM or L1 data cache *Memory Tagging Extension* (MTE) tag RAM.

### Configurations affected

This erratum affects configurations with CPU_CACHE_PROTECTION set.

### Conditions

This erratum occurs under the following conditions:

1. The core executes two or more stores to the same cache line but to different 16B-aligned quantities. At least one of the stores is a store of less than 32 bits, or is MTE tag-checked.
2. The partial or MTE tag-checked store above sees a correctable *Error Correcting Code* (ECC) error in the L1 data cache RAM or L1 data cache MTE tag RAM.
3. Very unlikely, timing-sensitive micro-architectural conditions occur.

### Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the conditions are met, a single-bit error could result in silent data corruption. This is similar to the case of a triple-bit error incorrectly being detected as a single-bit error.

One of the stores at condition (1) can write to the cache without marking the cache line as dirty. A second store that must already be in the store buffer will shortly mark the line dirty, but the line can be lost from the *Processing Element* (PE) caches in the small window between both operations. If the cache line leaves the PE caches in this window, the data from the store that has already been written to the cache might be lost, as it is not marked as dirty.

### Workaround

No workaround is required.