# Arm Cortex-A76AE (MP095)

# Software Developer Errata Notice

**Date of issue:** 31-Jan-2023

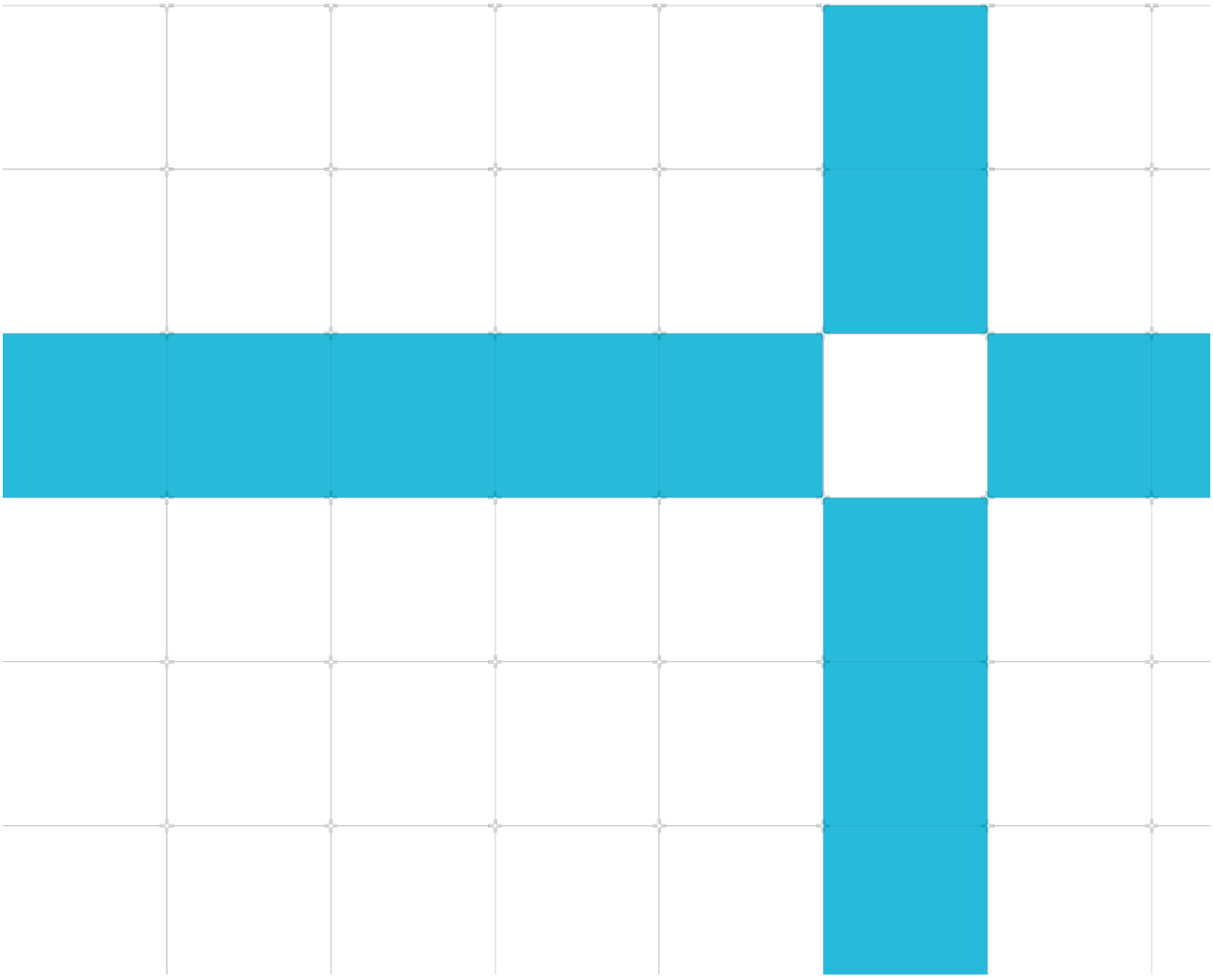
**Non-Confidential**

**Document version:** v15.0



**Document ID:** SDEN-1277541

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.

## Product status

The information in this document is for a product in development and is not final.

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Cortex-A76AE (MP095), 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

# 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.

31-Jan-2023: Changes in document version v15.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2816905 | New | Programmer | Category C | PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM |

14-Oct-2022: Changes in document version v14.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2753838 | New | Programmer | Category B | The core might deadlock during powerdown sequence |

31-Mar-2022: Changes in document version v13.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1931427 | Updated | Programmer | Category B | Atomic Store instructions to shareable write-back memory might cause memory consistency failures |
| 1931428 | Updated | Programmer | Category B | A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB |
| 1931431 | Updated | Programmer | Category B | Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR |
| 1931435 | Updated | Programmer | Category B | The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary |
| 1969399 | Updated | Programmer | Category B | External debugger access to Debug registers might not work during Warm reset |
| 1969401 | Updated | Programmer | Category B | Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics |
| 2371140 | Updated | Programmer | Category B | Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch |
| 1411009 | Updated | Programmer | Category C | TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0 |
| 1493281 | Updated | Programmer | Category C | Waypoints from previous session might cause single-shot comparator match when trace enabled |
| 1627786 | Updated | Programmer | Category C | ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect |
| 1931219 | Updated | Programmer | Category C | External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior |
| 1931423 | Updated | Programmer | Category C | Possible loss of CTI event |
| 1931424 | Updated | Programmer | Category C | Loss of CTI events during warm reset |

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1931430 | Updated | Programmer | Category C | Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR |
| 1931433 | Updated | Programmer | Category C | A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data |
| 1931437 | Updated | Programmer | Category C | ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect |
| 1931438 | Updated | Programmer | Category C | L2 data RAM may fail to report corrected ECC errors |
| 1931441 | Updated | Programmer | Category C | Uncorrectable tag errors in L2 cache might cause deadlock |
| 1931443 | Updated | Programmer | Category C | Noncompliance with prioritization of Exception Catch debug events |
| 1931445 | Updated | Programmer | Category C | PFG duplicate reported faults through a Warm reset |
| 1931448 | Updated | Programmer | Category C | Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO |
| 1969400 | Updated | Programmer | Category C | The PE might deadlock if Pseudofault Injection is enabled in Debug State |
| 2006577 | Updated | Programmer | Category C | DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL |
| 2033219 | Updated | Programmer | Category C | ETM trace information records a branch to the next instruction as an N atom |
| 2091838 | Updated | Programmer | Category C | An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers |
| 2110731 | Updated | Programmer | Category C | External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register |
| 2148888 | Updated | Programmer | Category C | A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely |
| 2266056 | Updated | Programmer | Category C | PMU L1D_CACHE_REFILL_OUTER is inaccurate |
| 2266059 | Updated | Programmer | Category C | Reads of DISR_EL1 incorrectly return 0s while in Debug State |
| 2266060 | Updated | Programmer | Category C | DRPS instruction is not treated as UNDEFINED at EL0 in Debug state |
| 2266063 | Updated | Programmer | Category C | L1 Data poison is not cleared by a store |
| 2346936 | Updated | Programmer | Category C | ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk |
| 2421293 | Updated | Programmer | Category C | Software-step not done after exit from Debug state with an illegal value in DSPSR |

08-Feb-2022: Changes in document version v12.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2371140 | New | Programmer | Category B | Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch |
| 2266056 | Updated | Programmer | Category C | PMU L1D_CACHE_REFILL_OUTER is inaccurate |
| 2346936 | New | Programmer | Category C | ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk |
| 2421293 | New | Programmer | Category C | Software-step not done after exit from Debug state with an illegal value in DSPSR |

16-Aug-2021: Changes in document version v11.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1931431 | Updated | Programmer | Category B | Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR |
| 2110731 | New | Programmer | Category C | External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register |
| 2148888 | New | Programmer | Category C | A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely |
| 2266056 | New | Programmer | Category C | PMU L1D_CACHE_REFILL_OUTER is inaccurate |
| 2266059 | New | Programmer | Category C | Reads of DISR_EL1 incorrectly return 0s while in Debug State |
| 2266060 | New | Programmer | Category C | DRPS instruction is not treated as UNDEFINED at EL0 in Debug state |
| 2266063 | New | Programmer | Category C | L1 Data poison is not cleared by a store |

02-Mar-2021: Changes in document version v10.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2033219 | New | Programmer | Category C | ETM trace information records a branch to the next instruction as an N atom |
| 2091838 | New | Programmer | Category C | An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers |

06-Nov-2020: Changes in document version v9.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2006577 | New | Programmer | Category C | DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL |

29-Sep-2020: Changes in document version v8.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1969399 | New | Programmer | Category B | External debugger access to Debug registers might not work during Warm reset |
| 1969401 | New | Programmer | Category B | Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics |
| 1969400 | New | Programmer | Category C | The PE might deadlock if Pseudofault Injection is enabled in Debug State |

11-Sep-2020: Changes in document version v7.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1931427 | New | Programmer | Category B | Atomic Store instructions to shareable write-back memory might cause memory consistency failures |
| 1931428 | New | Programmer | Category B | A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB |
| 1931431 | New | Programmer | Category B | Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR |
| 1931435 | New | Programmer | Category B | The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary |
| 1931219 | New | Programmer | Category C | External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior |
| 1931423 | New | Programmer | Category C | Possible loss of CTI event |
| 1931424 | New | Programmer | Category C | Loss of CTI events during warm reset |
| 1931430 | New | Programmer | Category C | Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR |
| 1931433 | New | Programmer | Category C | A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data |
| 1931437 | New | Programmer | Category C | ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect |
| 1931438 | New | Programmer | Category C | L2 data RAM may fail to report corrected ECC errors |
| 1931441 | New | Programmer | Category C | Uncorrectable tag errors in L2 cache might cause deadlock |
| 1931443 | New | Programmer | Category C | Noncompliance with prioritization of Exception Catch debug events |
| 1931445 | New | Programmer | Category C | PFG duplicate reported faults through a Warm reset |
| 1931448 | New | Programmer | Category C | Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO |

28-Feb-2020: Changes in document version v6.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1493250 | Updated | Programmer | Category B | Software Step might prevent interrupt recognition |
| 1450382 | Updated | Programmer | Category B (rare) | MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data |
| 1411009 | Updated | Programmer | Category C | TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0 |
| 1411012 | Updated | Programmer | Category C | Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError |
| 1411013 | Updated | Programmer | Category C | L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events |
| 1411016 | Updated | Programmer | Category C | Read from PMCCNTR in AArch32 might return corrupted data |
| 1450068 | Updated | Programmer | Category C | MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit |
| 1450069 | Updated | Programmer | Category C | Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address |
| 1450070 | Updated | Programmer | Category C | Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line |
| 1493246 | Updated | Programmer | Category C | Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences |
| 1493281 | Updated | Programmer | Category C | Waypoints from previous session might cause single-shot comparator match when trace enabled |
| 1506939 | Updated | Programmer | Category C | TRCIDR3.CCITMIN value is incorrect |
| 1539760 | Updated | Programmer | Category C | Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE |
| 1539763 | Updated | Programmer | Category C | CPUECTLR_EL1 controls for the MMU have no affect |
| 1627786 | New | Programmer | Category C | ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect |
| 1662733 | New | Programmer | Category C | MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption |
| 1683870 | New | Programmer | Category C | Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock |
| 1745148 | New | Programmer | Category C | APB access to trace registers does not work during Warm reset |

15-Nov-2019: Changes in document version v5.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1539760 | New | Programmer | Category C | Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE |
| 1539763 | New | Programmer | Category C | CPUECTLR_EL1 controls for the MMU have no affect |

28-Jun-2019: Changes in document version v4.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1493250 | New | Programmer | Category B | Software Step might prevent interrupt recognition |
| 1493246 | New | Programmer | Category C | Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences |
| 1493281 | New | Programmer | Category C | Waypoints from previous session might cause single-shot comparator match when trace enabled |
| 1506939 | New | Programmer | Category C | TRCIDR3.CCITMIN value is incorrect |

05-Apr-2019: Changes in document version v3.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1450382 | New | Programmer | Category B (rare) | MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data |
| 1450068 | New | Programmer | Category C | MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit |
| 1450069 | New | Programmer | Category C | Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address |
| 1450070 | New | Programmer | Category C | Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line |

12-Mar-2019: Changes in document version v2.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 1411009 | New | Programmer | Category C | TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0 |
| 1411012 | New | Programmer | Category C | Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError |
| 1411013 | New | Programmer | Category C | L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events |
| 1411016 | New | Programmer | Category C | Read from PMCCNTR in AArch32 might return corrupted data |

07-Dec-2018: Changes in document version v1.0

No errata in this document version.

# 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 |
|---|---|---|---|---|---|
| 1493250 | Programmer | Category B | Software Step might prevent interrupt recognition | r0p0 | r1p0 |
| 1931427 | Programmer | Category B | Atomic Store instructions to shareable write-back memory might cause memory consistency failures | r0p0, r1p0 | r1p1 |
| 1931428 | Programmer | Category B | A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB | r0p0, r1p0 | r1p1 |
| 1931431 | Programmer | Category B | Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR | r0p0, r1p0 | r1p1 |
| 1931435 | Programmer | Category B | The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary | r0p0, r1p0 | r1p1 |
| 1969399 | Programmer | Category B | External debugger access to Debug registers might not work during Warm reset | r0p0, r1p0 | r1p1 |
| 1969401 | Programmer | Category B | Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics | r0p0, r1p0 | r1p1 |
| 2371140 | Programmer | Category B | Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch | r0p0, r1p0, r1p1 | Open |
| 2753838 | Programmer | Category B | The core might deadlock during powerdown sequence | r0p0, r1p0, r1p1 | Open |
| 1450382 | Programmer | Category B (rare) | MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data | r0p0 | r1p0 |
| 1411009 | Programmer | Category C | TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0 | r0p0, r1p0, r1p1 | Open |
| 1411012 | Programmer | Category C | Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError | r0p0 | r1p0 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 1411013 | Programmer | Category C | L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events | r0p0 | r1p0 |
| 1411016 | Programmer | Category C | Read from PMCCNTR in AArch32 might return corrupted data | r0p0 | r1p0 |
| 1450068 | Programmer | Category C | MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit | r0p0 | r1p0 |
| 1450069 | Programmer | Category C | Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address | r0p0 | r1p0 |
| 1450070 | Programmer | Category C | Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line | r0p0 | r1p0 |
| 1493246 | Programmer | Category C | Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences | r0p0 | r1p0 |
| 1493281 | Programmer | Category C | Waypoints from previous session might cause single-shot comparator match when trace enabled | r0p0, r1p0 | r1p1 |
| 1506939 | Programmer | Category C | TRCIDR3.CCITMIN value is incorrect | r0p0 | r1p0 |
| 1539760 | Programmer | Category C | Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE | r0p0 | r1p0 |
| 1539763 | Programmer | Category C | CPUECTLR_EL1 controls for the MMU have no affect | r0p0 | r1p0 |
| 1627786 | Programmer | Category C | ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect | r0p0, r1p0 | r1p1 |
| 1662733 | Programmer | Category C | MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption | r0p0 | r1p0 |
| 1683870 | Programmer | Category C | Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock | r0p0 | r1p0 |
| 1745148 | Programmer | Category C | APB access to trace registers does not work during Warm reset | r0p0 | r1p0 |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 1931219 | Programmer | Category C | External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior | r0p0, r1p0 | r1p1 |
| 1931423 | Programmer | Category C | Possible loss of CTI event | r0p0, r1p0 | r1p1 |
| 1931424 | Programmer | Category C | Loss of CTI events during warm reset | r0p0, r1p0 | r1p1 |
| 1931430 | Programmer | Category C | Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR | r0p0, r1p0 | r1p1 |
| 1931433 | Programmer | Category C | A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data | r0p0, r1p0 | r1p1 |
| 1931437 | Programmer | Category C | ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect | r0p0, r1p0 | r1p1 |
| 1931438 | Programmer | Category C | L2 data RAM may fail to report corrected ECC errors | r0p0, r1p0 | r1p1 |
| 1931441 | Programmer | Category C | Uncorrectable tag errors in L2 cache might cause deadlock | r0p0, r1p0 | r1p1 |
| 1931443 | Programmer | Category C | Noncompliance with prioritization of Exception Catch debug events | r0p0, r1p0, r1p1 | Open |
| 1931445 | Programmer | Category C | PFG duplicate reported faults through a Warm reset | r0p0, r1p0 | r1p1 |
| 1931448 | Programmer | Category C | Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO | r0p0, r1p0 | r1p1 |
| 1969400 | Programmer | Category C | The PE might deadlock if Pseudofault Injection is enabled in Debug State | r0p0, r1p0 | r1p1 |
| 2006577 | Programmer | Category C | DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL | r0p0, r1p0, r1p1 | Open |
| 2033219 | Programmer | Category C | ETM trace information records a branch to the next instruction as an N atom | r0p0, r1p0, r1p1 | Open |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2091838 | Programmer | Category C | An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers | r0p0, r1p0, r1p1 | Open |
| 2110731 | Programmer | Category C | External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register | r0p0, r1p0, r1p1 | Open |
| 2148888 | Programmer | Category C | A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely | r0p0, r1p0, r1p1 | Open |
| 2266056 | Programmer | Category C | PMU L1D_CACHE_REFILL_OUTER is inaccurate | r0p0, r1p0, r1p1 | Open |
| 2266059 | Programmer | Category C | Reads of DISR_EL1 incorrectly return 0s while in Debug State | r0p0, r1p0, r1p1 | Open |
| 2266060 | Programmer | Category C | DRPS instruction is not treated as UNDEFINED at EL0 in Debug state | r0p0, r1p0 | r1p1 |
| 2266063 | Programmer | Category C | L1 Data poison is not cleared by a store | r0p0, r1p0, r1p1 | Open |
| 2346936 | Programmer | Category C | ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk | r0p0, r1p0, r1p1 | Open |
| 2421293 | Programmer | Category C | Software-step not done after exit from Debug state with an illegal value in DSPSR | r0p0, r1p0, r1p1 | Open |
| 2816905 | Programmer | Category C | PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM | r0p0, r1p0, r1p1 | Open |

# Errata descriptions

## Category A

There are no errata in this category.

## Category A (rare)

There are no errata in this category.

# Category B

## 1493250
## Software Step might prevent interrupt recognition

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r1p0.

### Description

The Software Stepping of a system call instruction (SVC, HVC, or SMC) can prevent recognition of subsequent interrupts when Software Stepping is disabled in the exception handler of the system call. Additionally, unconventional code involving the Software Stepping of an MSR instruction that clears the MDSCR_EL1.SS bit (disables Software Step while stepping) can prevent recognition of subsequent interrupts.

### Configurations Affected

This erratum affects all configurations.

### Conditions:

Case A:

1. Software Step is enabled.
2. The system configuration is (MDSCR_EL1.KDE==1) or (MDSCR_EL1.KDE==0 and HCR_EL2.E2H==1 and (HCR_EL2.TGE==1 or MDCR_EL2.TDE==1)).
3. An ERET with SPSR_ELx.SS==1 is executed to cause the Software Step state machine to enter the active-not-pending state.
4. A system call instruction (SVC, HVC, or SMC) is executed and generates its system call exception (that is, it is not trapped).
5. The exception handler of the system call disables Software Step by clearing MDSCR_EL1.SS or by setting SPSR_ELx.D such that, upon return, no Software Step exception is taken.

Case B:

1. Software Step is enabled.
2. An ERET with SPSR_ELx.SS==1 is executed to cause the Software Step state machine to enter the active-not-pending state.
3. An MSR MDSCR_EL1 instruction that clears the MDSCR_EL1.SS bit is executed (disables Software Step).

## Implications

Case A:

Arm believes that for this product, MDSCR_EL1.KDE is not set to 1 by deployed devices in the field and is only used when debugging the system software during initial product development. In these cases, the effect of the erratum is for interrupts to be disabled even after switching to other software contexts that are not being debugged as part of the system software debugging. Arm believes that a workaround does not need to be deployed for the situation where MDSCR_EL1.KDE==1, and a workaround is not available.

Some devices are expected to run an operating system at EL2 with HCR_EL2.E2H set to 1. The implication of this erratum for such a system is that single-stepping of an untrusted user application at EL0 can lead to subsequent execution not recognizing interrupts where it should, leading to errant behavior. The software workaround described below can be deployed in the operating system at EL2 to prevent single-stepping of untrusted user applications from triggering this erratum.

Case B:

Unconventional code involving the Software Stepping of the disabling instruction is not expected to be encountered, therefore no workaround is required.

## Workaround

When Software Step is used to debug an application under an operating system running at EL2 with HCR_EL2.E2H set to 1, the software workaround involves explicitly triggering a Software Step exception with modifications to the system call exception handler code and Software Step exception handler code. This entails setting MDSCR_EL1.KDE and MDSCR_EL1.SS and clearing PSTATE.D to trigger a Software Step exception from the system call handler. The Software Step handler then sets SPSR_ELx.D before returning back to the system call handler, where MDSCR_EL1.KDE and MDSCR_EL1.SS are restored to their original values.

If a workaround is required when MDSCR_EL1.KDE is set to 1, then please contact Arm.

# 1931427
# Atomic Store instructions to shareable write-back memory might cause memory consistency failures

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Atomic Store instructions to shareable write-back memory that are performed as far atomics might cause memory consistency failures if the initiating PE has a shared copy of the cache line containing the addressed memory.

## Configurations Affected

This erratum affects all configurations that have an interconnect capable of handling far atomic transactions indicated by the BROADCASTATOMIC pin being set to 1.

## Conditions

1. PE0 executes Atomic Store instruction that hits in the L1 data cache and L2 cache in the Shared state.
2. PE0 changes the L2 state to Invalid, sends an invalidating snoop to the L1 data cache, and issues a AtomicStore transaction on the CHI interconnect.
3. PE0 invalidating snoop to the L1 data cache is delayed due to internal queueing.

## Implications

If the above conditions are met, PE0 might not observe invalidating snoops caused by other PEs in the same coherency domain and thus might violate memory consistency for loads to the same cache line as the Atomic Store.

## Workaround

Set CPUACTLR2_EL1[2] to force Atomic Store operations to write-back memory to be performed in the L1 data cache.

# 1931428
# A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Under certain conditions, a transient single-bit ECC error in the MMU TC RAM might prevent a TLB invalidate (TLBI) instruction from removing the entry. If the transient error is not detected for a subsequent miss request targeting the affected page, then the MMU might return a stale translation.

## Configurations Affected

All configurations are affected.

## Conditions

All of the following conditions must be met:

- Both stage 1 and stage 2 translations are enabled.
- Stage 1 page or block size is larger than stage 2 page or block size.
- MMU TC RAM entry has a transient single-bit ECC error.
- TLBI targets the translation in the MMU TC RAM entry containing the single-bit ECC error.
- The single-bit ECC error prevents the TLBI from removing the entry.
- Transient single-bit ECC error goes away before a subsequent translation request matching the L2 TLB entry is issued.

## Implications

If the above conditions are met, then the MMU might return stale translation for a subsequent access. The transient single-bit ECC error will be reported in ERR0MISC0_EL1 register.

## Workaround

This condition can be detected by ERR0STATUS[25:24] == 0b10, indicating a corrected error, and ERR0MISC0[3:0] == 0b0010, indicating the source to be the MMU TC RAM. If enabled, the fault handling interrupt is asserted when an error is recorded in the ERR0* error record, and software can check for this condition. Software should treat this condition as an uncontainable uncorrected error (i.e. as if ERR0STATUS[29,21:20] == 0b100).

# 1931431
# Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

If a load or store crosses a cache line (cache line size = 64 bytes) and a watchpoint address targets a location in the upper cache line, the Fault Address Register (FAR) or the External Debug Watchpoint Address Register (EDWAR) (if set up for Debug Halt) will contain an incorrect address.

## Configurations Affected

This erratum affects all configurations.

## Conditions

Incorrect address in FAR or EDWAR appears when the:

1. Watchpoint targets a double word (or less or more) at cache line address B.
2. Load or store targets accesses two cache lines: lower cache line A and upper cache line B. The cache line size is 64 bytes.

## Implications

FAR contains the target address of load or store.

EDWAR contains the target address of load or store if enabled for Debug Halt.

## Workaround

There is no hardware workaround.

The following software workaround can be applied:

If the Fault Address Register (FAR) or External Debug Watchpoint Address Register (EDWAR) does not match a watchpoint, software can attempt to identify a relevant watchpoint:

a) For A DC ZVA whose address is not aligned to DCZID_EL0.BS, by rounding the faulting address down to a cache line boundary (64 bytes) and attempting to match this against active watchpoints.

Note: Most software aligns addresses used by DC ZVA, and this case is expected to be rare in practice.

b) For all other loads and stores, by attempting to use the address of the next cache line boundary (64 bytes) and attempting to match this against active watchpoints.

# 1931435
# The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

When the core executes a conditional branch instruction with software step or halt step, the core might write an incorrect address into ELR_ELn after the core completes stepping.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is stepping a conditional branch instruction located at the end of a 32-byte aligned block.
2. The conditional branch is resolved as not taken.

## Implications

If the above conditions are met, the core might not write the correct instruction address (PC+4 of stepping instruction) into the ELR_ELn register after stepping is completed.

## Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1, which delays instruction fetch after branch misprediction. This workaround will have a small impact on performance.

# 1969399
# External debugger access to Debug registers might not work during Warm reset

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

During Warm reset, external debugger access for Debug registers might be ignored.

## Configurations Affected

All configurations are affected.

## Conditions

1. Warm reset is asserted.
2. External debugger access is initiated for one of following Debug registers:
   a. DBGBCR<n>_EL1 (n=0-5)
   b. DBGBVR<n>_EL1 (n=0-5)
   c. EDECCR

## Implications

If the above conditions are met, the core might ignore the access request. The read operation might return incorrect data. The write operation might not take effect and stale data might be retained.

## Workaround

There is no workaround.

# 1969401
# Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Under certain conditions, atomic instructions with acquire semantics might not be ordered with respect to older instructions with release semantics. The older instruction could either be a store or store atomic.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Load atomic, CAS or SWP with acquire but no release semantics is executed.
2. There is an older instruction with release semantics and it could either be a store to non-WB memory or a store atomic instruction that is executed as a far atomic.

## Implications

If the above condition are met, a memory ordering violation might happen.

## Workaround

This erratum can be avoided by inserting a DMB ST before acquire atomic instructions without release semantics. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers:

```
LDR x0,=0x3
MSR S3_6_c15_c8_0,x0
LDR x0,= 0x10E3900002
MSR S3_6_c15_c8_2,x0
LDR x0,= 0x10FFF00083
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x2001003FF
MSR S3_6_c15_c8_1,x0
```

```
LDR x0,=0x4
MSR S3_6_c15_c8_0,x0
LDR x0,= 0x10E3800082
MSR S3_6_c15_c8_2,x0
LDR x0,= 0x10FFF00083
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x2001003FF
MSR S3_6_c15_c8_1,x0

LDR x0,=0x5
MSR S3_6_c15_c8_0,x0
LDR x0,= 0x10E3800200
MSR S3_6_c15_c8_2,x0
LDR x0,= 0x10FFF003E0
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x2001003FF
MSR S3_6_c15_c8_1,x0

ISB
```

# 2371140
# Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0 and r1p1. Open.

## Description

A PE executing a PLDW or PRFM PST instruction that lies on a mispredicted branch path might cause a second PE executing a store exclusive to the same cache line address to fail continuously.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. One PE is executing store exclusive.
2. A second PE has branches that are consistently mispredicted.
3. The second PE instruction stream contains a PLDW or PRFM PST instruction on the mispredicted path that accesses the same cache line address as the store exclusive executed by the first PE.
4. PLDW/PRFM PST causes an invalidation of the first PE's caches and a loss of the exclusive monitor.

## Implications

If the above conditions are met, the store exclusive instruction might continuously fail.

## Workaround

Set CPUACTLR2_EL1[0] to 1 to force PLDW/PFRM ST to behave like PLD/PRFM LD and not cause invalidations to other PE caches. There might be a small performance degradation to this workaround for certain workloads that share data.

# 2753838
# The core might deadlock during powerdown sequence

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

While powering down the Processing Element (PE), a correctable L2 tag ECC error might cause a deadlock in the power-down sequence.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. Error detection and correction is enabled through ERXCTLR_EL1.ED=1.
2. PE executes more than 24 writes to Device-nGnRnE or Device-nGnRE memory.
3. PE executes power-down sequence as described in TRM.

## Implications

If the above conditions are met, the PE might deadlock during the hardware cache flush that automatically occurs as part of the power-down sequence.

## Workaround

Add a DSB instruction before the ISB of the power-down code sequence specified in the TRM.

# Category B (rare)

## 1450382
## MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data

### Status

Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0. Fixed in r1p0.

### Description

An MRRC read of certain Generic Timer system registers in AArch32 mode might return corrupt data.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs when the following conditions are met under rare internal timing conditions:

1. The core is executing at AArch32 at EL0.
2. An MRRC to CNTPCT, CNTVCT, CNTP_CVAL, or CNTV_CVAL is executed.

### Implications

If the erratum occurs, then the second destination register [Rt2] of the MRRC will incorrectly contain the same data as the first destination register [Rt].

### Workarounds

The erratum can be avoided by trapping MRC/MCR/MRRC/MCRR accesses in AArch32 to the affected registers and doing the equivalent code sequence in the trap handler.
To trap the CNT* accesses, set CNTKCTL_EL1.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0. If HCR_EL2.{E2H,TGE}={1,1} then set CNTHCTL_EL2.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0.
The following registers will be trapped:

- CNTP_CTL.
- CNTP_CVAL.
- CNTP_TVAL.

- CNTV_CTL.
- CNTV_CVAL.
- CNTV_TVAL.
- CNTPCT.
- CNTVCT.
- CNTFRQ.

# Category C

## 1411009
## TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

### Description

TLBI instructions are not treating ASID[15:8] as zero when TCR_EL1.AS=0, as specified in the Arm Architecture Reference Manual. In this configuration, the bits are RES0, which should be written to zero by software, and ignored by hardware.

### Configurations Affected

The erratum affects all configurations.

### Conditions

1. TCR_EL1.AS=0.
2. A TLBI is executed with ASID[15:8] not equal to zero.

### Implications

The TLBI will execute locally and broadcast with an ASID that is out of range for this configuration.

### Workaround

This erratum can be avoided if software is properly writing zero to RES0 bits.

# 1411012
# Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

When an Uncontainable (UC) SError is reported or deferred by the core, it might be incorrectly logged as an Unrecoverable (UEU) SError. This is an inappropriate categorization downgrade which might allow for silent error propagation.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. An Uncontainable (UC) SError occurs in the system.
2. The Uncontainable (UC) SError is reported or deferred.

## Implications

If the above conditions are met, then the ESR_ELx.AET or DISR_EL1.AET field might log the Uncontainable (UC) SError as an Unrecoverable (UEU) SError.

## Workaround

This erratum can be mitigated by treating all SErrors reported with type Unrecoverable (UEU) as type Uncontainable (UC).

# 1411013
# L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

The L1D_CACHE access related PMU events 0x4, 0x40, and 0x41 and the L1D_TLB access related PMU events 0x25, 0x4E, and 0x4F are incorrectly counting non-memory read/write operations that must be excluded. Software prefetch instructions are counted as read accesses and all other instructions are counted as write accesses.

## Configurations Affected

This erratum affects all configurations.

## Conditions

A software prefetch (PRFM) instruction or one of the following non-memory write operations is issued to the Load/Store Unit:

- A barrier (DMB, DSB, ESB, or PSB).
- A TLB Maintenance Operation (TMO).
- A Cache Maintenance Operation (CMO).
- An Address Translation operation (AT).
- A debug RAM read operation.

## Implications

If any of the non-memory read/write operations listed above are issued to the Load/Store Unit, then the PMU counts for events L1D_CACHE (0x4), L1D_CACHE_RD (0x40), L1D_CACHE_WR (0x41) or L1D_TLB (0x25), L1D_TLB_RD (0x4E), and L1D_TLB_WR (0x4F) are incremented incorrectly.

## Workaround

There is no workaround for this erratum.

# 1411016
# Read from PMCCNTR in AArch32 might return corrupted data

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

When PMCCNTR is configured to count core clock cycles, the result of a read from the PMCCNTR system register in AArch32 state might be corrupted. This corruption is predictable and occurs when the clock cycle count rolls over into the upper 32 bits of the register. For example, if PMCCNTR=0xFFFF_FFFF and a read is executed around the time the clock cycle count is incremented, then the value returned might be 0x1_FFFF_FFFF rather than 0x1_0000_0000.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. PMCCNTR is configured to count core clock cycles.
2. The lower 32 bits of PMCCNTR contains a value close to 0xFFFF_FFFF.
3. A read from PMCCNTR is performed in AArch32.

## Implications

If the above conditions are met, then the read from the PMCCNTR register might return corrupted data.

## Workaround

This erratum is not expected to require a workaround.

# 1450068
# MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

An MSR DSPSR_EL0 instruction that is executed in debug state and alters the Debug Saved Program Status Register, might fail to update PSTATE.{N,Z,C,V,GE} values on exit from debug state. This erratum applies to both AArch32 (MCR DSPSR) and AArch64 (MSR DSPSR_EL0) operation.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is in debug state.
2. The core executes an MSR instruction to alter the Debug Saved Program Status Register.
3. The core exits debug state.
4. The core might expose the incorrect PSTATE through execution of a conditional instruction or a read of PSTATE.{N,Z,C,V,GE} state.

## Implications

If the above conditions are met, then this erratum might result in data corruption, incorrect program flow, or produce other undesirable effects. However, this erratum will not result in violation of access controls, for example, this erratum will not result in the core making accesses to Secure memory from Non-secure mode.

## Workaround

The erratum can be avoided by setting CPUACTLR_EL1[45] to 1 prior to exiting from debug state. Power consumption in the core will be higher when CPUACTLR_EL1[45] is 1, as this prevents dynamic clock gating within sections of the core.

# 1450069
# Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

The errant behavior described in this erratum pertains solely to ETM reporting information, and strictly in close vicinity of ETM reporting of an indirect branch with a malformed branch target address (a programming error).

Information recorded in the ETM trace buffer for branch instructions includes the Virtual Address (VA) of the branch target. An indirect branch has a malformed branch target address when either the lowermost bits of the target address stipulate a misaligned instruction address, or the uppermost bits are non-canonical. Execution of an indirect branch with a malformed target address results in an Instruction Abort. ETM trace information correctly reports the malformed target address for the branch execution, and also correctly reports exception information for the Instruction Abort.
However, under rare circumstances, a few branches immediately preceding the indirect branch with malformed target address can incorrectly include the upper and lower portions of the malformed target address in the ETM trace information for the target of these earlier branches. Only the upper and lower portions of the branch target VAs are potentially mis-reported in the ETM trace information.

## Configurations Affected

This erratum affects all configurations.

## Conditions:

1. ETM is enabled.
2. An indirect branch with a malformed branch target address is executed and traced.
3. Branch instructions immediately preceding the indirect branch with malformed target address are executed and traced.

## Implications

If the above conditions are met, then within a tightly constrained window the branches immediately preceding the indirect branch with malformed target address might record partially corrupted target addresses in the ETM trace buffer.

## Workaround

No workaround is required. The programming error should be evident to users from the ETM trace information pertaining to the indirect branch with a malformed branch target address and trace information from its resultant Instruction Abort.

# 1450070
# Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

If a core detects a false miss due to a single bit L1 tag RAM ECC error when executing a younger load instruction that bypassed another load or barrier and completed by forwarding data from a prior store instruction, then an ordering violation might occur in the presence of a snoop request.

## Configurations Affected

The erratum affects all multi-core configurations with CORE_CACHE_PROTECTION= 1.

## Conditions

1. Core A has a cache line X resident in the L1 data cache with write permissions, and has one or more stores in flight.
2. Core A performs a load (LD1) out-of-order for line X, bypassing another load or a barrier. The load encounters a tag single-bit ECC error, which makes the line appear as a miss, it allocates a miss request buffer requesting the line from L2.
3. LD1 is able to complete by forwarding data from an older store.
4. The older store drains and updates the L1 data cache.
5. Core B sends a snoop for line X and the snoop is ordered ahead of the miss request from LD1.
6. Core B performs a store to the line X.
7. Core A then receives the line X on behalf of its read request from LD1 and allocates the line.
8. Core A does not detect an ordering violation for the following:
    - An older load LD2 now observes this newer store, or
    - LD1 bypassed a load with acquire or barrier and is now required to observe the newer store.

## Implications

If the above conditions are met, then under specific microarchitectural timing conditions, there might be an ordering violation, such as a read after read violation.

This has been graded as Programmer Category C because Arm expects this erratum to have a negligible impact over the undetected ECC failure rate in real systems. The reason for this categorization is that the issue only occurs during a very short window in time when using a highly implausible code sequence involving racing writes by multiple different cores.

## Workaround

There is no workaround for this erratum.

# 1493246
# Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

If a write to the External Debug Registers occurs such that it activates an address breakpoint, then under certain conditions the core might stop executing a few instructions before the breakpoint exception should occur.

## Configurations Affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

## Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The breakpoint is set on a cacheable line.
3. The breakpoint is set on a cache line that starts with the final 16 bits of a 32-bit instruction.
4. There is a stuck-at-fault in the L1 instruction data array near the breakpoint location.
5. The breakpoint is activated using the External Debug Registers while the core is fetching.

## Implications

If the above conditions are met, then the core might deadlock.

## Workaround

Any interrupt will break the core out of the deadlock state.

# 1493281
# Waypoints from previous session might cause single-shot comparator match when trace enabled

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

On the first waypoint after the core ETM is enabled, it is possible for a single-shot comparator to have a spurious match based on the address from the last waypoint in the previous trace session.

## Configurations Affected

This erratum affects all configurations.

## Conditions

- The core ETM has been enabled, disabled, and re-enabled since the last reset.
- Single-shot address comparators are enabled.
- The last waypoint address before the core ETM was disabled either matches a single-shot comparator or causes a match in the range between waypoints depending on the single-shot control setup.

## Implications

There might be a spurious single-shot comparator match, which might be used by the trace analyzer to activate other trace events.

## Workaround

Between tracing sessions, set the core ETM to enter a prohibited region either instead of or in addition to disabling the ETM.

# 1506939
# TRCIDR3.CCITMIN value is incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

Software reads of the TRCIDR3.CCITMIN field, corresponding to the instruction trace counting minimum threshold, observe the value 0x100 or a minimum cycle count threshold of 256. The correct value should be 0x4 for a minimum cycle count threshold of 4.

## Configurations Affected

This erratum affects all configurations.

## Conditions

- Software reads the TRCIDR3 ID register.
- Software uses the value of the CCITMIN field to determine minimum instruction trace cycle counting threshold to program the ETM.

## Implications

If software uses the value returned by the TRCIDR3.CCITMIN field, then it will limit the range which could be used for programming the ETM. In reality, the ETM could be programmed with a much smaller value than what is indicated by the TRCIDR3.CCITMIN field and function correctly.

## Workaround

The value for the TRCIDR3.CCITMIN field should be treated as 0x4.

# 1539760
# Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

If a Virtual SError is pending and masked at the current Exception level when an ESB instruction is executed, then the VDISR_EL2 update occurs properly but in some cases the clearing of HCR_EL2.VSE might not occur. This failure to clear HCR_EL2.VSE can only occur when the Virtual SError is masked.

## Configurations Affected

This erratum affects all configurations.

## Conditions:

1. A Virtual SError is pending at the current Exception level.
2. Virtual SErrors are masked at the current Exception level.
3. An ESB instruction executes.

## Implications

If the above conditions are met, then under specific microarchitectural timing conditions HCR_EL2.VSE might not be cleared to 0, which is required by the Arm architecture. This might result in spurious Virtual SErrors. Under all circumstances, the Virtual SError syndrome from VSESR_EL2 is correctly recorded in VDISR_EL2 and VDISR_EL2.A is correctly set to 1.

## Workaround

A workaround is not expected to be required. This is because existing software only executes ESB instructions at EL2 and above. If your software executes ESB instructions at EL1 with the conditions described above, then contact Arm support for more details.

# 1539763
# CPUECTLR_EL1 controls for the MMU have no affect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

The CPUECTLR_EL1 register contains IMPLEMENTATION DEFINED configuration and control options for the MMU. The MMU bits affected by this erratum are CPUECTLR_EL1[54:46]. Any changes to these values have no affect on the functionality or performance.

## Configurations Affected

This erratum affects all configurations.

## Conditions

Software updates to modify MMU control bits CPUECTLR_EL1[54:46] from reset values have no effect.

## Implications

Software attempts to change the functionality or performance of the core by changing reset values of CPUECTLR_EL1[54:46] have no affect. The value is updated in the register correctly, such that any subsequent read of the CPUECTLR_EL1 register will return the expected data, however, the modifications have no affect on the behavior of the core.

## Workaround

There is no workaround.

# 1627786
# ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Under certain conditions, the ERR0MISC0_EL1.SUBARRAY value recorded for ECC errors in the L1 data cache might be incorrect.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. A load, store, or atomic instruction accesses multiple banks of the L1 data cache.
2. One of the banks accessed has an ECC error.

## Implications

If the above conditions are met, then ERR0MISC0_EL1.SUBARRAY might have an incorrect value. The remaining fields of the ERR0MISC0_EL1 register remain correct.

## Workaround

There is no workaround for this erratum.

# 1662733
# MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

In AArch32, MRC reads of DBGDSCRint into destination APSR_nzcv (Rt=15) always produce a result of 0. Also, if there is a younger MRC or MRRC read to any accessible register following the DBGDSCRint read into APSR_nzcv, then the younger read result might be corrupted.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is in AArch32 state at EL0.
2. An MRC read of DBGDSCRint into APSR_nzcv (Rt=15) occurs.

## Implications

If the above conditions are met, then:

1. APSR_nzcv is always written with 0.
2. Under specific microarchitectural timing conditions in AArch32 EL0, a subsequent MRC or MRRC might be corrupted.

## Workaround

Directly read DBGDSCRint with an MRC instruction into a general-purpose register (R0-R14), and then write that general-purpose register to the flags by doing an MSR APSR_f. To avoid the possible corruption, add an ISB instruction before any subsequent MRC or MRRC instructions.

# 1683870
# Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

Under certain conditions, executing a cache maintenance by set/way instruction targeting the L1 data cache in close proximity to multiple snoops where the older snoop detects a transient ECC error might result in a deadlock.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core has executed at least two snoop requests looking up the L1 data cache. These could have been generated internally from this core or from another core in the system.
2. The older snoop detects a transient single-bit or double-bit ECC error, but at least two snoops have performed a lookup of the L1 data cache.
3. The core executes a cache maintenance by set/way instruction targeting the L1 data cache.
4. The snoops are required to perform another lookup due to the ECC error detected. All snoops are rescheduled to maintain ordering of the snoop transactions.
5. The snoop transactions continuously retry the L1 data cache lookup, preventing the cache maintenance operation from completing.

## Implications

If the above conditions are met under certain timing conditions, then the snoops might not make progress, resulting in a deadlock. Arm does not expect cache maintenance operations by set/way to be executed in most code sequences, since hardware mechanisms have been incorporated for flushing the caches as a part of powerdown sequences. Software is expected to use cache maintenance operations by VA to manage coherency.

Note that cache maintenance by set/way instructions are UNDEFINED at EL0.

## Workaround

Software should avoid the use of cache maintenance operations by set/way. A hypervisor should trap these instructions by setting HCR_EL2.TSW = 1 and emulate the instructions with equivalent cache maintenance operations by virtual address for the entire address space of the guest.

# 1745148
# APB access to trace registers does not work during Warm reset

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r1p0.

## Description

During Warm reset, APB writes to trace registers are ignored, and reads return incorrect data. Trace continues through Warm reset over the ATB interface as expected.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Warm reset is asserted.
2. Trace registers are accessed over the APB interface.

## Implications

If the above conditions are met, then APB writes to the trace registers are ignored. APB reads to the trace registers return incorrect data.

## Workaround

The workaround for this erratum is to set up the trace registers in the needed configuration before entering Warm reset.

# 1931219
# External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

In Debug state with SCTLR_ELx.IESB set to 1, memory uploads and downloads executed in memory access mode might lead to unpredictable behavior for the current exception level.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Core is In Debug state.
2. SCTLR_ELx.IESB is set to 1 for the current exception level.
3. Memory access mode is enabled via EDSCR.MA set to 1.

## Implications

If the above conditions are met, memory upload and download behavior is unpredictable for the current exception level and might lead to incorrect operation or results. The unpredictable behavior is limited to legal behavior at the current exception level.

## Workaround

The erratum can be avoided by clearing SCTLR_ELx.IESB before performing memory uploads or downloads in Debug state using memory access mode.

# 1931423
# Possible loss of CTI event

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

A CTI event from the core to the external DebugBlock might be dropped, in rare occurrences, if close in temporal proximity to a previous CTI event.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. CTI event occurs.
2. Another CTI event occurs before completion of the processing of the previous CTI event.

## Implications

CTI events might be dropped.

## Workaround

This erratum has no workaround.

# 1931424
# Loss of CTI events during warm reset

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

ETM external output CTI events from the core to the external DebugBlock will not be reported during warm reset.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. An ETM external output CTI event occurs while warm reset is asserted.

## Implications

The ETM external output CTI event will be dropped and any cross triggering that depends on this CTI event will not occur. For example, if the ETM external output was to be used to trigger a trace capture component to stop trace capture, then trace capture will not stop due to this event.

## Workaround

This erratum has no workaround.

# 1931430
# Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

If the watchpoint address targets a lower portion of a cache line, but not all of the cache line, and the address target of the Data Cache Zero by VA (DC ZVA) falls in the upper portion of the cache line that the watchpoint does not target, the Fault Address Register (FAR) (or External Debug Watchpoint Address Register (EDWAR) if setup for Debug Halt) will contain an incorrect address.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Watchpoint targets double word (or less or more) at address A.
2. DC ZVA targets address greater than A+7, but less than A+63. The cache line size is 64 bytes, which is a mis-aligned address.

## Implications:

FAR contains target address of DC ZVA.
EDWAR contains target address of DC ZVA if enabled for Debug Halt.

## Workaround:

There is no hardware workaround. The common case for DC ZVA targets is to be granule aligned, thus most software will not be affected by this case.

# 1931433
# A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

A memory mapped write to PMSSRR at offset 0x6f4 might configure the Cycle counter and/or Performance Monitor event counters to be reset along with reset of corresponding overflow status bits in the PMOVSR register. The register supports read/write functionality instead of RAZ/WI.

## Configurations affected

This erratum affects all configurations.

## Conditions

1. System enables PMU snapshot mechanism.
2. System performs memory mapped write of PMSSRR setting PMSSRR[x], where x is 31 or any value from 0 to 5 (inclusive).
3. Snapshot trigger is seen through a legal mechanism.

## Implications

If the above conditions are met, the corresponding counter (PMCCNTR_EL0 if x=31 or PMEVCNTR<x>_EL0 if x = [0,5]) will reset after a snapshot is taken. Further, the corresponding bit in the PMOVSR_EL0 register will be reset.

A memory mapped read will return data that is written to these bits and 0 otherwise.

This register is supposed to have RAZ/WI functionality and no effect on other counters.

## Workaround

Avoid write of PMSSRR when system is using the PMU Snapshot mechanism.

# 1931437
# ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Under certain conditions, the ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values recorded for ECC errors in the L1 data cache might be incorrect.

## Configurations affected

This erratum affects configurations with CORE_CACHE_PROTECTION set to TRUE.

## Conditions

1. The L1 data cache contains both a single-bit and double-bit ECC error on different words within the same 64-byte cacheline.
2. An access is made to the cacheline in the L1 data cache containing both the single-bit and double-bit ECC errors simultaneously.

## Implications

If the above conditions are met, then ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE might have an incorrect values.

## Workaround

There is no workaround for this erratum.

# 1931438
# L2 data RAM may fail to report corrected ECC errors

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

For specific operation types and cache states, a read of the L2 data RAM might fail to report a detected and corrected single-bit ECC error.

## Configurations Affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

## Conditions

1. PE L1 data cache and L2 cache are in a SharedClean state and the exclusive monitor is armed for a given physical address.
2. PE executes a store exclusive instruction to this physical address.
3. L2 cache reads its data RAMs, and detects and corrects a single-bit ECC error.

## Implications

If the above conditions are met, the PE will correct the error, but might fail to report it in the RAS error log registers. This can cause a small loss in diagnostic capability.

## Workaround

There is no workaround.

# 1931441
# Uncorrectable tag errors in L2 cache might cause deadlock

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Under rare conditions that include the aliasing of multiple virtual addresses to a single physical address, a detected and reported double-bit ECC error in the L2 cache tag RAM might lead to a state in which an unexpected L1 cache eviction can cause a deadlock in the L2 cache.

## Configurations affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

## Conditions

1. L2 cache detects and reports a tag double-bit ECC error.
2. A set of rare conditions occur within the PE memory system.

## Implications

If the above conditions are met, the L2 transaction queue might deadlock and never complete the prefetch operation.

## Workaround

There is no workaround for this erratum.

# 1931443
# Noncompliance with prioritization of Exception Catch debug events

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

ARMv8.2 architecture requires that Debug state entry due to an Exception Catch debug event (generated on exception entry) occur before any asynchronous exception is taken at the first instruction in the exception handler. An asynchronous exception might be taken as a higher priority exception than Exception Catch and the Exception Catch might be missed altogether.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Debug Halting is allowed.
2. EDECCR bits are configured to catch exception entry to ELx.
3. A first exception is taken resulting in entry to ELx.
4. A second, asynchronous exception becomes visible at the same time as exception entry to ELx.
5. The second, asynchronous exception targets an Exception level ELy that is higher than ELx.

## Implications

If the above conditions are met, the core might recognize the second exception and not enter Debug state as a result of Exception Catch on the first exception. When the handler for the second exception completes, software might return to execute the first exception handler, and assuming the core does not halt for any other reason, the first exception handler will be executed and entry to Debug state via Exception Catch will not occur.

## Workaround

When setting Exception Catch on exceptions taken to an Exception level ELx, the debugger should do either or both of the following:

1. Ensure that Exception Catch is also set for exceptions taken to all higher Exception Levels, so that the second (asynchronous) exception generates an Exception Catch debug event.
2. Set Exception Catch for an Exception Return to ELx, so that when the second (asynchronous)

exception handler completes, the exception return to ELx generates an Exception Catch debug event.

Additionally, when a debugger detects that the core has halted on an Exception Catch to an Exception level ELy, where y > x, it should check the ELR_ELy and SPSR_ELy values to determine whether the exception was taken on an ELx exception vector address, meaning an Exception Catch on entry to ELx has been missed.

# 1931445
# PFG duplicate reported faults through a Warm reset

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

Under certain conditions, the Pseudo-fault Generation Error Record Registers might generate duplicate faults through a Warm reset.

## Configurations affected

All configurations are affected.

## Conditions

1. ERR0PFGCDN is set with a non-zero countdown value.
2. ERR0PFGCTL is set to generate a pseudo-fault with ERR0PFGCTL.CDEN enabled.
3. The countdown value expires, generating a pseudo-fault.
4. Warm reset asserts.

## Implications

After the Warm reset, a second generated pseudo-fault might occur.

## Workaround

De-assert the ERR0PFGCTL control bits before asserting a Warm reset.

# 1931448
# Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

If a Corrected Error is recorded because of a bus error which has no valid location (ERR0STATUS.MV=0x0), then a subsequent Corrected Error might incorrectly increment either of the ERR0MISC0.CECR or ERR0MISC0.CECO counters.

## Configurations Affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

## Conditions

1. A Corrected Error which has no valid location (ERR0STATUS.MV=0x0) is recorded.
2. A subsequent Corrected Error occurs.

## Implications

The subsequent Corrected Error might improperly increment either of the ERR0MISC0.CECR or ERR0MISC0.CECO counters.

## Workaround

No workaround is expected to be required.

# 1969400
# The PE might deadlock if Pseudofault Injection is enabled in Debug State

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

If Pseudofault Injection is enabled for the PE node (ERR0PFGCTL.CDNEN=0x1) and the PE subsequently enters Debug State, then the PE might deadlock. Alternatively, if the PE is executing in Debug State and the PE enables Pseudofault Injection for the PE node (ERR0PFGCTL.CDNEN=0x1), then the PE might deadlock.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. ERR0PFGCTL.CDNEN is set to 0x1 to enable Pseudofault Injection.
2. The PE enters Debug State.

OR

1. The PE is executing in Debug State.
2. ERR0PFGCTL.CDNEN is set to 0x1 to enable Pseudofault Injection.
   Implications
   If the above conditions are met, then the PE might deadlock.

Workaround
Ensure ERR0PFGCTL.CDNEN=0x0 before entering Debug State and while executing in Debug State.

# 2006577
# DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

In Debug state with SCTLR_ELx.IESB set to 1, the DRPS (debug only) instruction does not execute properly. Only partial functionality of the DRPS instruction is performed.

## Configurations Affected

This erratum affects all configurations.

## Conditions

The erratum occurs under the following conditions:

1. The core is in Debug state.
2. SCTLR_ELx.IESB is set to 1 for the current exception level.
3. The DRPS instruction is executed.

## Implications

If the above conditions are met, the DRPS instruction does not complete as intended, which might lead to incorrect operation or results. Register data or memory will not be corrupted. There are also no security or privilege violations.

## Workaround

The erratum can be avoided by clearing SCTLR_ELx.IESB followed by the insertion of an ISB and an ESB instruction in code before the DRPS instruction.

# 2033219
# ETM trace information records a branch to the next instruction as an N atom

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

If a branch is taken to the next instruction, and if the instruction state remains the same, then the ETM traces it as an N atom rather than an E atom or branch address packet. This is incorrect as the ETM architecture says a taken branch should be traced as an E atom. This affects all forms of branches. State-changing branches are traced correctly.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This issue might occur when:

1. ETM is enabled.
2. A branch is taken to the next instruction.
3. The instruction state does not change.

## Implications

A trace decoder that interprets an N atom to move to the next instruction in the same state without a push or pop from the return stack will correctly maintain the control flow but will not be able to infer anything from a conditional branch.

A trace decoder that checks if unconditional branches were not traced as N atom might report an error.

## Workaround

To ensure continued control flow, ensure the trace decoder always interprets an N atom to move to the next instruction in same state without a push or pop from the return stack.

# 2091838
# An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

When an **MSR** instruction and an APB write operation are processed on the same cycle, the **MSR** instruction might not update the destination register correctly.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. A CPU executes an **MSR** instruction to update any of following SPR registers:
   a. DBGBCR<n>_EL1
   b. DBGBVR<n>_EL1
   c. DBGWCR<n>_EL1
   d. DBGWVR<n>_EL1
   e. OSECCR_EL1
2. An external debugger initiates an APB write operation for any of following registers:
   a. DBGBCR<n>
   b. DBGBVR<n>
   c. DBGBXVR<n>
   d. DBGWCR<n>
   e. DBGWVR<n>
   f. DBGWXVR<n>
   g. EDECCR
   h. EDITR
3. The SPR registers (for example, OSLSR_EL1.OSLK and EDSCR.TDA) and external pins are programmed to allow the following behavior:
   a. The execution of an **MSR** instruction in condition 1 to update its destination register without neither a system trap nor a debug halt
   b. The APB write operation in condition 2 to update its destination register without error
4. The **MSR** instruction execution in condition 1 and APB write operation in condition 2 happen in same

cycle.
5. The **MSR** write and the APB write are to two different registers. The architecture specifies that it is the software or debugger's responsibility to ensure writes to the same register are updated as expected.

## Implications

If the above conditions are met, an execution of the **MSR** instruction might not update the destination register correctly. The destination register might contain one of following values after execution:

1. The execution of the **MSR** instruction is ignored. The destination register of the **MSR** instruction holds an old value.
2. The execution of the **MSR** instruction writes an incorrect value to its destination register.

A external debugger and system software are expected to be coordinated to prevent conflict in these registers.

## Workaround

No workaround is required for this erratum.

# 2110731
# External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

The core might incorrectly issue a write to External Debug Instruction Transfer Register (EDITR) when an external APB write to another register that is located at offset 0x084 is performed in the Debug state. The following debug components share the offset alias with the EDITR register:

- ETE - TRCVIIECTLR - ViewInst Include/Exclude Control Register
- Reserved locations

The following debug component shares the offset alias with the EDITR register when the PE is configured with 20-PMUs:

- PMU - PMEVCNTR16[63:32] - Event Counter 16

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The core is in debug state.
2. The External Debug Status and Control Register (EDSCR) cumulative error flag field is 0b0.
3. Memory access mode is disabled, in example, EDSCR.MA = 0b0.
4. The OS Lock is unlocked.
5. External APB write is performed to a memory mapped register at offset 0x084 other than the EDITR.

## Implications

If the above conditions are met, then the core might issue a write to the EDITR and try to execute the instruction pointed to by the ITR. As a result of the execution, the following might happen:

- CPU state and/or memory might get corrupted.
- The CPU might generate an UNDEFINED exception.
- The EDSCR.ITE bit will be set to 0.

## Workaround

Before programming any register at this offset when the PE is in Debug state, the debugger should either:

- Set the EDSCR.ERR bit by executing some Undefined instruction (e.g. writing zero to EDITR); or
- Set the OS Lock and then unlock it afterwards.

# 2148888
# A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

Executing an A64 WFI or WFE instruction while in Debug state results in suspension of execution, and execution cannot be resumed by the normal WFI or WFE wake-up events while in Debug state.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The Processing Element (PE) is in Debug state and in AArch64 Execution state.
2. A WFI or WFE instruction is executed from EDITR.

## Implications

If the above conditions are met, the PE will suspend execution.

This is not thought to be a serious erratum, because an attempt to execute a WFI or WFE instruction while in Debug state is not expected.

For WFI executed in Debug state, execution can only resume by any of the following:

- A Cold or Warm reset
- A Restart request trigger event from the Cross Trigger Interface (CTI) causing exit from Debug state, followed by a WFI wake-up event

For WFE executed in Debug state, execution can only resume by any of the following:

- A Cold or Warm reset
- A Restart request trigger event from the CTI causing exit from Debug state, followed by a WFE wake-up event
- An external event that sets the Event Register. Examples include executing an SEV instruction on another PE in the system or an event triggered by the Generic Timer.

## Workaround

A workaround is unnecessary, because an attempt to execute a WFI or WFE instruction while in Debug state is not expected.

# 2266056
# PMU L1D_CACHE_REFILL_OUTER is inaccurate

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

The L1D_CACHE_REFILL_OUTER PMU event 0x45 is inaccurate due to ignoring refills generated from a system cache. The L1D_CACHE_REFILL PMU event 0x3 should be the sum of PMU events L1D_CACHE_REFILL_INNER 0x44 and L1D_CACHE_REFILL_OUTER 0x45, however, due to the inaccuracy of L1D_CACHE_REFILL_OUTER 0x45 it is possible that this might not be the case.

Note: L1D_CACHE_REFILL PMU event 0x3 does accurately count all L1D cache refills, including refills from a system cache.

## Configurations Affected

This erratum affects all configurations which implement a system cache.

## Conditions

This erratum occurs under the following conditions:

1. The L2 inner cache is allocated with data transferred from a system cache.

## Implications

When the previous condition is met, the L1D_CACHE_REFILL_OUTER PMU event 0x45 does not increment properly.

## Workaround

The correct value of L1D_CACHE_REFILL_OUTER PMU event 0x45 can be calculated by subtracting the value of L1D_CACHE_REFILL_INNER PMU event 0x44 from L1D_CACHE_REFILL PMU event 0x3.

# 2266059
# Reads of DISR_EL1 incorrectly return 0s while in Debug State

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

When the Processing Element (PE) is in Debug State, reads of DISR_EL1 from EL1 or EL2 with SCR_EL3.EA=0x1 will incorrectly return 0s.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The PE is executing in Debug State at EL1 or EL2, with SCR_EL3.EA=0x1.
2. The PE executes an MRS to DISR_EL1.

## Implications

If the above conditions are met, then the read of DISR_EL1 will incorrectly return 0s.

## Workaround

No workaround is expected to be required.

# 2266060
# DRPS instruction is not treated as UNDEFINED at EL0 in Debug state

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

## Description

In Debug state, DRPS is not treated as an UNDEFINED instruction.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The Processing Element (PE) is in Debug state.
2. PE is executing at EL0.
3. PE executes DRPS instruction.

## Implications

If the above conditions are met, then the PE will incorrectly execute DRPS as NOP instead of treating it as an UNDEFINFED instruction.

## Workaround

There is no workaround.

# 2266063
# L1 Data poison is not cleared by a store

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

The L1 Data poison is not cleared by a store under certain conditions.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. A Processing Element (PE) executes a store that does not write a full word to a location that has data marked as poison.
2. The PE executes another store that writes to all bytes that contain data poison before the previous store is globally observable.

## Implications

If the above conditions are met, then the poison bit in the L1 Data cache does not get cleared.

## Workaround

This erratum can be avoided by inserting a DMB before and after a word-aligned store that is intended to clear the poison bit.

# 2346936
# ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

When a data double bit error or external abort is encountered during a translation table walk, a synchronous exception is reported with the ISV bit set in the ESR_ELx register.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following condition:

1. A data double bit error or external abort is encountered during a translation table walk, and a synchronous exception is reported.

## Implications

If the previous condition is met, the ESR_ELx.ISV bit will be set. The ESR[23:14] bits are set with the correct syndrome for the instruction making the access. That is SAS, SSE, SRT, SF, and AR are all set according to the instruction.

## Workaround

This erratum has no workaround.

# 2421293
# Software-step not done after exit from Debug state with an illegal value in DSPSR

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

On exit from Debug state, PSTATE.SS is set according to DSPSR.SS and DSPSR.M.
If DSPSR.M encodes an illegal value, then PSTATE.SS should be set according to the current Exception level. When the erratum occurs, the PE always writes PSTATE.SS to 0.

## Configurations Affected

This erratum affects all configurations.

## Conditions

- Software-step is enabled in current Exception level
- DSPSR.M encodes an illegal value, like:
  - M[4] set
  - M is a higher Exception level than current Exception level
  - M targets EL2 or EL1, when they are not available
- DSPSR.D is not set
- DSPSR.SS is set

## Implications

If the previous conditions are met, then, on exit from Debug state the PE will directly take a Software-step Exception, without stepping an instruction as expected from DSPSR.SS=1.

## Workaround

This erratum has no workaround.

# 2816905
# PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r1p1. Open.

## Description

Under certain conditions, the *Processing Element* (PE) might fail to report multiple uncorrectable *Error Correction Code* (ECC) errors that occur in the L1 data cache tag RAM.

## Configurations affected

This erratum affects all configurations.

## Conditions

1. The PE detects and reports an uncorrectable ECC error in the L1 data cache tag RAM.
2. The PE detects a second uncorrectable ECC error in the L1 data cache tag RAM and an uncorrectable ECC error in the L1 data cache data RAM.

## Implications

If the previous conditions are met, then the PE might fail to report the second uncorrectable ECC error in the L1 data cache tag RAM and the address recorded in ERR0ADDR might have an incorrect value. The ECC error occurring in the L1 data cache data RAM is reported correctly.

## Workaround

No workaround is necessary. This erratum represents a condition where multiple uncorrectable ECC errors occur in a short period of time. While the PE does not report the errors correctly, ECC still provides a valuable mechanism for error detection and correction.