# Arm Cortex-A78C (MP154)

# Software Developer Errata Notice

**Date of issue:** 22-Feb-2023

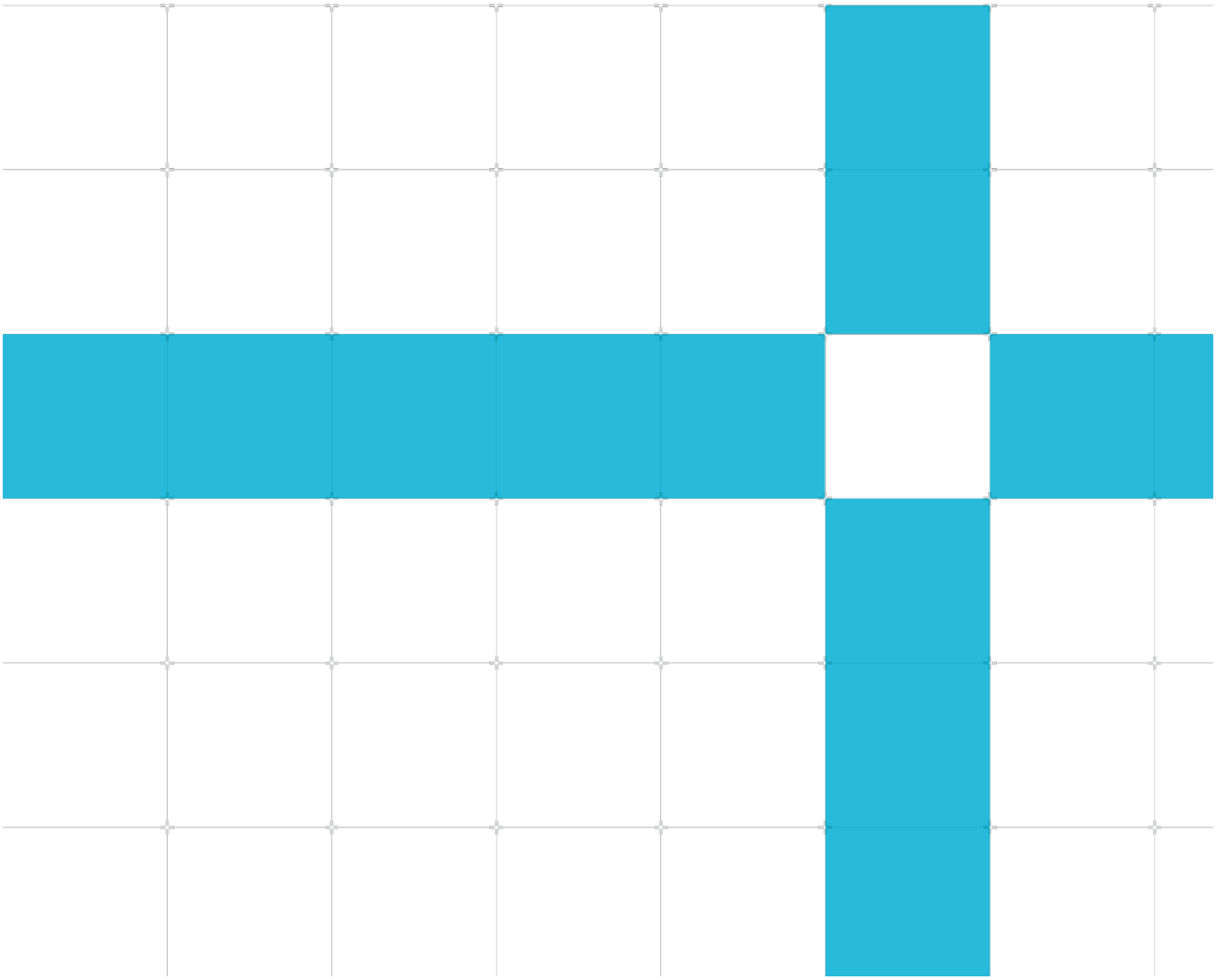
**Non-Confidential**

**Document version:** v8.0



**Document ID:** SDEN-2004089

This document contains all known errata since the r0p1 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-A78C (MP154), 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.

22-Feb-2023: Changes in document version v8.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2820249 | New | Programmer | Category C | PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM |

09-Nov-2022: Changes in document version v7.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2743232 | New | Programmer | Category B | Page crossing access that generates an MMU fault on the second page could result in a livelock |
| 2772122 | New | Programmer | Category B | The core might deadlock during powerdown sequence |
| 2779484 | New | Programmer | Category B | The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation |
| 2764620 | New | Programmer | Category C | Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP |

04-Aug-2022: Changes in document version v6.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2683027 | New | Programmer | Category B | Pointer Authentication controls might become corrupt |
| 2712575 | New | Programmer | Category B | Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back |
| 2242642 | New | Programmer | Category C | An SError might not be reported for an atomic store that encounters data poison |
| 2280363 | New | Programmer | Category C | PMU L1D_CACHE_REFILL_OUTER is inaccurate |
| 2446532 | New | Programmer | Category C | PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly |
| 2699194 | New | Programmer | Category C | Incorrect value reported for SPE PMU event SAMPLE_FEED |
| 2699200 | New | Programmer | Category C | Reads of DISR_EL1 incorrectly return 0s while in Debug State |
| 2699764 | New | Programmer | Category C | Incorrect read value for Performance Monitors Control Register |
| 2708636 | New | Programmer | Category C | DRPS instruction is not treated as UNDEFINED at EL0 in Debug state |
| 2712568 | New | Programmer | Category C | Incorrect read value for Performance Monitors Configuration Register EX field |

21-Jan-2022: Changes in document version v5.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2376749 | New | Programmer | Category B | Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch |
| 2395411 | New | Programmer | Category B | Translation table walk folding into an L1 prefetch might cause data corruption |
| 2341666 | New | Programmer | Category C | ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk |
| 2346736 | New | Programmer | Category C | Lower priority exception might be reported when abort condition is detected at both stages of translation |
| 2423051 | New | Programmer | Category C | Software-step not done after exit from Debug state with an illegal value in DSPSR |

24-Sep-2021: Changes in document version v4.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2242638 | New | Programmer | Category B | PDP deadlock due to CMP/CMN + B.AL/B.NV fusion |
| 2296019 | New | Programmer | Category C | L1 Data poison is not cleared by a store |

13-May-2021: Changes in document version v3.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2132064 | New | Programmer | Category B | Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock |
| 2005130 | Updated | Programmer | Category C | DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL |
| 2091747 | Updated | Programmer | Category C | CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation |
| 2102460 | Updated | Programmer | Category C | ETM trace information records a branch to the next instruction as an N atom |
| 2102761 | 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 |
| 2131909 | New | Programmer | Category C | Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes |
| 2132046 | New | Programmer | Category C | OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain |
| 2151899 | New | Programmer | Category C | A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely |

03-Mar-2021: Changes in document version v2.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2091747 | New | Programmer | Category C | CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation |
| 2102460 | New | Programmer | Category C | ETM trace information records a branch to the next instruction as an N atom |
| 2102761 | 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 |
| 2106995 | 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 |

02-Nov-2020: Changes in document version v1.0

| ID | Status | Area | Category | Summary |
|---|---|---|---|---|
| 2004081 | New | Programmer | Category C | Noncompliance with prioritization of Exception Catch debug events |
| 2004085 | New | Programmer | Category C | IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB |
| 2004088 | New | Programmer | Category C | The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified |
| 2005130 | New | Programmer | Category C | DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL |

# 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 |
|---|---|---|---|---|---|
| 2132064 | Programmer | Category B | Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock | r0p1, r0p2 | Open |
| 2242638 | Programmer | Category B | PDP deadlock due to CMP/CMN + B.AL/B.NV fusion | r0p1, r0p2 | Open |
| 2376749 | Programmer | Category B | Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch | r0p1, r0p2 | Open |
| 2395411 | Programmer | Category B | Translation table walk folding into an L1 prefetch might cause data corruption | r0p1, r0p2 | Open |
| 2683027 | Programmer | Category B | Pointer Authentication controls might become corrupt | r0p1, r0p2 | Open |
| 2712575 | Programmer | Category B | Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back | r0p1, r0p2 | Open |
| 2743232 | Programmer | Category B | Page crossing access that generates an MMU fault on the second page could result in a livelock | r0p1, r0p2 | Open |
| 2772122 | Programmer | Category B | The core might deadlock during powerdown sequence | r0p1, r0p2 | Open |
| 2779484 | Programmer | Category B | The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation | r0p1, r0p2 | Open |
| 2004081 | Programmer | Category C | Noncompliance with prioritization of Exception Catch debug events | r0p1, r0p2 | Open |
| 2004085 | Programmer | Category C | IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB | r0p1, r0p2 | Open |
| 2004088 | Programmer | Category C | The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified | r0p1, r0p2 | Open |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2005130 | Programmer | Category C | DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL | r0p1 | r0p2 |
| 2091747 | Programmer | Category C | CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation | r0p1 | r0p2 |
| 2102460 | Programmer | Category C | ETM trace information records a branch to the next instruction as an N atom | r0p1 | r0p2 |
| 2102761 | Programmer | Category C | External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register | r0p1 | r0p2 |
| 2106995 | 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 | r0p1, r0p2 | Open |
| 2131909 | Programmer | Category C | Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes | r0p1 | r0p2 |
| 2132046 | Programmer | Category C | OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain | r0p1 | r0p2 |
| 2151899 | Programmer | Category C | A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely | r0p1, r0p2 | Open |
| 2242642 | Programmer | Category C | An SError might not be reported for an atomic store that encounters data poison | r0p1, r0p2 | Open |
| 2280363 | Programmer | Category C | PMU L1D_CACHE_REFILL_OUTER is inaccurate | r0p1, r0p2 | Open |
| 2296019 | Programmer | Category C | L1 Data poison is not cleared by a store | r0p1, r0p2 | Open |
| 2341666 | Programmer | Category C | ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk | r0p1, r0p2 | Open |
| 2346736 | Programmer | Category C | Lower priority exception might be reported when abort condition is detected at both stages of translation | r0p1, r0p2 | Open |
| 2423051 | Programmer | Category C | Software-step not done after exit from Debug state with an illegal value in DSPSR | r0p1, r0p2 | Open |

| ID | Area | Category | Summary | Found in versions | Fixed in version |
|---|---|---|---|---|---|
| 2446532 | Programmer | Category C | PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly | r0p1, r0p2 | Open |
| 2699194 | Programmer | Category C | Incorrect value reported for SPE PMU event SAMPLE_FEED | r0p1, r0p2 | Open |
| 2699200 | Programmer | Category C | Reads of DISR_EL1 incorrectly return 0s while in Debug State | r0p1, r0p2 | Open |
| 2699764 | Programmer | Category C | Incorrect read value for Performance Monitors Control Register | r0p1, r0p2 | Open |
| 2708636 | Programmer | Category C | DRPS instruction is not treated as UNDEFINED at EL0 in Debug state | r0p1, r0p2 | Open |
| 2712568 | Programmer | Category C | Incorrect read value for Performance Monitors Configuration Register EX field | r0p1, r0p2 | Open |
| 2764620 | Programmer | Category C | Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP | r0p1, r0p2 | Open |
| 2820249 | Programmer | Category C | PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM | r0p1, r0p2 | Open |

# Errata descriptions

## Category A

There are no errata in this category.

## Category A (rare)

There are no errata in this category.

# Category B

## 2132064
## Disabling of data prefetcher with outstanding prefetch TLB miss may cause a hang

### Status

Fault Type: Programmer Category B
Fault Status: Present in r0p1, r0p2. Open.

### Description

If the data prefetcher is disabled (by an MSR to CPUECTLR register) while a prefetch TLB miss is outstanding, the processor may hang on the next context switch.

### Configurations Affected

All configurations are affected.

### Conditions

- MSR write to CPUECTLR register that disables the data prefetcher.
- A TLB miss from the prefetch TLB is outstanding.

### Implications

If the above conditions are met, a hang may occur on the next context switch.

### Workaround

- Workaround option 1:
  If the following code surrounds the MSR, it will prevent the erratum from happening:
  - tlbi (to blind address) local version (does not have to be broadcast)
  - dsb
  - isb
  - MSR CPUECTLR - disabling the prefetcher
  - isb

- Workaround option 2:
  Place the data prefetcher in the most conservative mode instead of disabling it. This will greatly reduce prefetches but not eliminate them. This is accomplished by writing the following bits to the

value indicated:
- ecltr[7:6], PF_MODE = 2'b11

# 2242638
# PDP deadlock due to CMP/CMN + B.AL/B.NV fusion

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p1, r0p2. Open.

## Description

When Performance Defined Power (PDP) is enabled, a Compare (CMP) or Compare negative (CMN) instruction followed by a conditional branch of form B.AL or B.NV might cause a deadlock.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. PDP configuration is enabled.
2. Execution of CMP/CMN, followed by B.AL/B.NV.

## Implications

If above conditions are met, then a deadlock might result, requiring a reset of the processor.

## Workaround

This erratum can be avoided by applying following patch. These instructions are not expected to be present in the code often, so any performance impact should be minimal. The code sequence should be applied early in the boot sequence prior to any of the possible errata conditions being met.

```
LDR x0,=0x5
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0x10F600E000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0x10FF80E000
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x80000000003FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0

ISB
```

# 2376749
# 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 r0p1, r0p2. 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.

# 2395411
# Translation table walk folding into an L1 prefetch might cause data corruption

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p1, and r0p2. Open

## Description

A translation table walk that matches an existing L1 prefetch with a read request outstanding on CHI might fold into the prefetch, which might lead to data corruption for a future instruction fetch.

## Configurations Affected

This erratum affects all configurations

## Conditions

1. In specific microarchitectural situations, the PE merges a translation table walk request with an older hardware or software prefetch L2 cache miss request.

## Implications

If the previous conditions are met, an unrelated instruction fetch might observe incorrect data.

## Workaround

Disable folding of demand requests into older prefetches with L2 miss requests outstanding by setting CPUACTLR2_EL1[40] to 1.

# 2683027
# Pointer Authentication controls might become corrupt

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p1, r0p2. Open.

## Description

When the Processing Element (PE) writes to any of the following Pointer Authentication control bits, the state of the bit might become corrupt:

- SCR_EL3.API
- HCR_EL2.API
- SCTLR_EL3.(ENDB,ENDA,ENIB,ENIA)
- SCTLR_EL2.(ENDB,ENDA,ENIB,ENIA)
- SCTLR_EL1.(ENDB,ENDA,ENIB,ENIA)

This can result in Pointer Authentication related instructions exhibiting unexpected behaviors, for example improperly executing as a NOP or failing to correctly TRAP:

- AUTDA, AUTDB, AUTDZA, AUTDZB, AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZA, AUTIZB
- PACGA, PACDA, PACDB, PACDZA, PACDZB, PACIA, PACIA1716, PACIASP, PACIAZ, PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZA, PACIZB
- RETAA, RETAB, BRAA, BRAB, BLRAA, BLRAB, BRAAZ, BRABZ, BLRAAZ, BLRABZ
- ERETAA, ERETAB, LDRAA, and LDRAB

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. The PE writes to any of the listed Pointer Authentication control bits.
2. The PE executes a Pointer Authentication related instruction.

## Implications

If the above conditions are met, then the Pointer Authentication instruction might exhibit unexpected behaviors.

## Workaround

This erratum can be avoided by flushing the mop cache following a write to registers SCR_EL3, HCR_EL2, or SCTLR_ELx. This can be implemented through execution of the following code at EL3 as soon as possible after boot:

```
    LDR x0,=0x3
    MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
    LDR x0,=0xEE010F10
    MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
    LDR x0,=0xFF1F0FFE
    MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
    LDR x0,=0x100000004003FF
    MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
    ISB
```

# 2712575
# The core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back

## Status

Fault Type: Programmer Category B
Fault Status: Present in r0p1, r0p2. Open.

## Description

If a core is fetching instructions from memory while stage 1 translation is disabled and instruction cache is disabled, the core ignores Stage 2 forced Write-Back indication programmed by HCR_EL2.FWB and make Non-cacheable, Normal memory request. This may cause the core to fetch stale data from memory subsystem.

## Configurations Affected

This erratum might affect system configurations that do not use Arm interconnect IP.

## Conditions

The erratum occurs if all the following conditions apply:

- The *Processing Element* (PE) is using EL1 translation regime.
- Stage 2 translation is enabled (HCR_EL2.VM=1).
- Stage 1 translation is disabled (SCTLR_EL1.M=0).
- Instruction cache is enabled from EL2 (HCR_EL2.ID=0).
- Instruction cache is disabled from EL1 (SCTLR_EL1.I=0).

## Implications

If the conditions are satisfied, the core makes all instruction fetch request as Non-cacheable, Normal memory regardless of stage 2 translation output even if Stage 2 Forced Write-back is enabled. This might cause the core to fetch stale data from memory because Non-cacheable memory access does not probe any of cache hierarchy (e.g., Level-2 cache). If the bypassed cache hierarchy contains data modified by other initiators, stale data might be fetched from memory.

## Workaround

For Hypervisor, initiating appropriate cache maintenance operations as if the core does not support stage 2 Forced Write-back feature. The cache maintenance operation should be initiated when new memory is allocated to a guest OS. This operation writeback the modified data in intermediate caches to point of coherency.

# 2743232
# Page crossing access that generates an MMU fault on the second page could result in a livelock

## Status

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

## Description

Under unusual micro-architectural conditions, a page crossing access that generates a *Memory Management Unit* (MMU) fault on the second page can result in a livelock.

## Configurations Affected

All configurations are affected.

## Conditions

This erratum occurs under all of the following conditions:

1. Page crossing load or store misses in the *Translation Lookaside Buffer* (TLB) and needs a translation table walk for both pages.
2. The table walk for the second page results in an MMU fault.

## Implications

If the above conditions are met, under unusual micro-architectural conditions with just the right timing, the core could enter a livelock. This is expected to be very rare and even a slight perturbation due to external events like snoops could get the core out of livelock.

## Workaround

This erratum can be avoided by setting CPUACTLR5_EL1[56:55] to 2'b01.

# 2772122
# The core might deadlock during powerdown sequence

## Status

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

## Description

While powering down the *Processing Element* (PE), a correctable L2 tag ECC error might cause a deadlock in the powerdown 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 powerdown sequence as described in the *Technical Reference Manual* (TRM).

## Implications

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

## Workaround

Add a DSB instruction before the ISB of the powerdown code sequence specified in the TRM.

## 2779484
## The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation

### Status

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

### Description

The Processing Element (PE) might generate memory accesses using invalidated mappings after completion of a Distributed Virtual Memory (DVM) SYNC operation.

### Configurations Affected

All configurations are affected.

### Conditions

This erratum can occur on a PE (PE0) only if the affected TLBI and subsequent DVM sync operations are broadcast from another PE (PE1). The TLBI and DVM sync operations executed locally by PE0 are not affected.

### Implications

When this erratum occurs, after completion of a DVM SYNC operation, the PE can continue generating memory accesses through mappings that were invalidated by a previous TLBI operation.

### Workaround

The erratum can be avoided by setting CPUACTLR3_EL1[47]. Setting this chicken bit might have a small impact on power and negligible impact on performance.

# Category B (rare)

There are no errata in this category.

# Category C

## 2004081
## Noncompliance with prioritization of Exception Catch debug events

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. 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.

# 2004085
# IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

After implementation-defined RAMINDEX register is programmed to initiate direct memory access to internal memory for Instruction TLB, implementation-defined IDATAn_EL3 value represents unpredictable value.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Implementation-defined RAMINDEX register is programmed to initiate direct memory access to internal memory for Instruction TLB.

## Implications

If the above conditions are met, IDATAn_EL3 register might represent incorrect value for Translation regime, VMID, ASID, and VA[48:21].

## Workaround

There is no workaround.

# 2004088
# The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

When a core fetches an instruction from a virtual address that is associated with a page table entry which has been modified and the fetched block is affected by parity error, the core might report an incorrect address within the same 32B block onto the Fault Address Register (FAR).

## Configurations Affected

All configurations are affected.

## Conditions

1. The core fetches instructions from an aligned 32B virtual address block.
2. A page table entry associated with the above 32B aligned block is updated. The new translation would cause an instruction abort.
3. TLB holds the old translation since the synchronization process, for example, TLB Invalidate (TLBI) followed by Data Synchronization Barrier (DSB), was not completed.
4. Some of the fetched instructions are affected by parity error in I-cache data RAM.
5. Context synchronization events were not processed between the last executed instruction and the above instruction.

## Implications

When above conditions are satisfied, a core might report an incorrect fetch address to FAR_ELx. The address reported in FAR_ELx points at an earlier location in the same aligned 32B block. FAR_ELx[63:5] still points correct virtual address.

## Workaround

There is no workaround.

# 2005130
# 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 r0p1. Fixed in r0p2.

## 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.

# 2091747
# CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation

## Status

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

## Description

When an instruction fetch is initiated for a page programmed as non-cacheable normal memory in stage-1 translation and as device memory in stage-2 translation, the instruction memory might incorrectly return 0. This might cause an unexpected UNDEFINED exception.

## Configurations Affected

The erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. A CPU fetch instruction from a page satisfies the following:
    - Stage-1 translation of this page is programmed as non-cacheable normal memory.
    - Stage-2 translation of this page is programmed as device memory.

## Implications

If the above conditions are met, the CPU might read 0 from the instruction memory. This instruction might cause an unexpected UNDEFINED exception. Instruction fetches to device memory are not architecturally predictable in any case, and device memory is expected to be marked as execute never, so this erratum is not expected to cause any problems to real-world software.

## Workaround

This erratum has no workaround.

# 2102460
# ETM trace information records a branch to the next instruction as an N atom

## Status

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

## 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.

# 2102761
# 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 r0p1. Fixed in r0p2.

## 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

## 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.

# 2106995
# 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 r0p1, r0p2. 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.

# 2131909
# Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes

## Status

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

## Description

Collision information captured by PMBSR_EL1.COLL might be lost under certain circumstances, when the buffer management interrupt is raised.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. A sampling collision event is detected.
2. Subsequent SPE write results in 2 SEI errors.

## Implications

If the above conditions are met, the collision indicator in PMBSR_EL1 is incorrectly set to 0, following the 2nd SEI error. PMBSR_EL1 does capture and set the "Data Loss" (DL) indicator and all the other PMBSR_EL1 fields correctly.

## Workaround

There is no workaround for this erratum.

# 2132046
# OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain

## Status

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

## Description

OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain. If a Warm Reset occurs, then the value in this register will be lost.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Warm Reset is asserted.

## Implications

If the above conditions are met, then the value in OSECCR_EL1/EDECCR will be lost.

## Workaround

A debugger should enable a Reset Catch debug event by setting EDECR.RCE to 1. This causes the PE to generate a Reset Catch debug event on a Warm reset, allowing the debugger to reprogram the EDECCR.

# 2151899
# A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. 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.

# 2242642
# An SError might not be reported for an atomic store that encounters data poison

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

Under certain conditions, an atomic store that encounters data poison might not report an SError.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs under the following conditions:

1. An atomic store that is unaligned to its data size but within a 16-byte boundary accesses memory.
2. The atomic store accesses multiple L1 data banks such that not all banks have data poison.

## Implications

If the above conditions are met, an SError might not be reported although poisoned data is consumed. Note that the data remains poisoned in the L1 and will be reported on the next access.

## Workaround

This erratum has no workaround.

# 2280363
# PMU L1D_CACHE_REFILL_OUTER is inaccurate

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. 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.

# 2296019
# L1 Data poison is not cleared by a store

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. 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.

# 2341666
# 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 r0p1, r0p2. 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.

## 2346736
## Lower priority exception might be reported when abort condition is detected at both stages of translation

### Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

### Description

When a permission fault or unsupported atomic fault is detected in the second stage of translation during stage 1 translation table walk, and there is a higher priority alignment fault due to SCTLR_EL1.C bit not being set, then Data Abort might be generated reflecting the lower priority fault.

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs when all the following conditions apply:

1. The core executes an atomic, load/store exclusive, or load-acquire/store-release instruction.
2. SCTLR_EL1.C bit is not set and access is not aligned to size of data element.
3. A permission fault or unsupported atomic fault is detected in the second stage of translation during stage 1 translation table walk.

### Implications

If the previous conditions are met, a Data Abort exception will be generated and incorrectly routed to EL2 with Data Fault Status Code (DFSC) of permission fault or unsupported atomic fault, when it should have been routed to EL1 with DFSC of alignment fault.

### Workaround

This erratum has no workaround.

# 2423051
# 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 r0p1, r0p2. 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.

# 2446532
# PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

The following Performance Monitoring Unit (PMU) events do not count correctly:

- 0x3D, STALL_SLOT_BACKEND, no operation sent for execution on a slot due to the backend
- 0x3E, STALL_SLOT_FRONTEND, no operation sent for execution on a slot due to the frontend

## Configurations Affected

This erratum affects all configurations.

## Conditions

One of the PMU event counters is configured to count any of the following events:

- 0x3D, STALL_SLOT_BACKEND
- 0x3E, STALL_SLOT_FRONTEND

## Implications

When operations are stalled in the processing element's dispatch pipeline slot, some of those slot stalls are counted as frontend stalls when they should have been counted as backend stalls, rendering PMU events 0x3D (STALL_SLOT_BACKEND) and 0x3E (STALL_SLOT_FRONTEND) inaccurate. The PMU event 0x3F (STALL_SLOT) does still accurately reflect its intended count of "No operation sent for execution on a slot".

## Workaround

This erratum has no workaround.

# 2699194
# Incorrect value reported for SPE PMU event SAMPLE_FEED

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

Under certain conditions when a CMP instruction is followed by a Branch, the SAMPLE_FEED PMU event 0x4001 is not reported.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. *Statistical Profiling Extension* (SPE) sampling is enabled.
2. SPE samples a CMP instruction, which is followed immediately by a BR instruction.

## Implications

If the above conditions are met, then the SAMPLE_FEED event may not be incremented.

For most expected use cases, the inaccuracy is not expected to be significant.

## Workaround

There is no workaround.

# 2699200
# Reads of DISR_EL1 incorrectly return 0s while in Debug State

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. 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.

# 2699764
# Incorrect read value for Performance Monitors Control Register

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

The Performance Monitors Control Register (PMCR_EL0) and the External Performance Monitor Control Register (PMCR) might return an incorrect read value for the X field.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. Software writes a nonzero value to the PMCR_EL0.X, or debugger writes a nonzero value to the PMCR.X
2. Software reads the PMCR_EL0 register, or debugger reads the PMCR register

## Implications

The PMCR_EL1.X or PMCR.X field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

## Workaround

This erratum has no workaround.

# 2708636
# DRPS instruction is not treated as UNDEFINED at EL0 in Debug state

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## 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.

# 2712568
# Incorrect read value for Performance Monitors Configuration Register EX field

## Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1, r0p2. Open.

## Description

The Performance Monitors Configuration Register (PMCFGR) might return an incorrect read value for the EX field.

## Configurations Affected

This erratum affects all configurations.

## Conditions

This erratum occurs when the software reads the PMCFGR register.

## Implications

The PMCFGR.EX field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

## Workaround

This erratum has no workaround.

# 2764620
# Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

## Status

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

## Description

Under certain conditions the SAMPLE_POP PMU event 0x4000 might continue to count after SPE profiling has been disabled.

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. *Statistical Profiling Extension* (SPE) sampling is enabled.
2. *Performance Monitoring Unit* (PMU) event counting is enabled.
3. SPE buffer is disabled, either directly by software, or indirectly via assertion of PMBIRQ, or by entry into Debug state.

## Implications

If the previous conditions are met, then the SAMPLE_POP event might reflect an overcounted value. The impact of this erratum is expected to be very minor for actual use cases, as SPE sampling analysis is typically performed independently from PMU event counting.

## Workaround

If a workaround is desired, then minimization of potential overcounting of the SAMPLE_POP event can be realized via software disable of any PMU SAMPLE_POP event counters whenever SPE is disabled, and also upon the servicing of a PMBIRQ interrupt. For profiling of EL0 workloads, software can further reduce exposure to overcounting by configuring the counter to not count at Exception levels of EL1 or higher.

# 2820249
# 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 r0p1, and r0p2. 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.