Arm® Cortex-X4 (MP161)

Software Developer Errata Notice

Date of issue: 07-Aug-2023

Non-Confidential

Document version: 6.0

Document ID: SDEN-2432808

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

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-X4 (MP161), 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.

07-Aug-2023: Changes in document version v6.0

ID	Status	Area	Category	Summary
2923985	New	Programmer	Category B	Branch prediction history not suppressed when switching from low to high EL
2959655	New	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly
3022725	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
2919943	New	Programmer	Category B (rare)	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level
2982002	New	Programmer	Category C	SPE latency counters are corrupted under certain conditions

14-Jun-2023: Changes in document version v5.0

ID	Status	Area	Category	Summary
2897503	New	Programmer	Category B	A load that passes a 32-byte store instruction in MTE precise mode which crosses a page boundary might violate memory ordering
2908612	New	Programmer	Category B	Incorrect value of PMSELR register resulting in Unknown Value.
2957258	New	Programmer	Category B	Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1
2794916	New	Programmer	Category C	DGH instruction doesn't execute correctly
2901622	New	Programmer	Category C	PMU event MEM_ACCESS_CHECKED_WR incorrectly counts aborted or inactive stores in MTE precise mode
2904903	New	Programmer	Category C	Incorrect event count for event 0x80c1 (Non-scalable FP element operations speculatively executed) in PMU
2910960	New	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts
2914109	New	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption
2928370	New	Programmer	Category C	PE might report an unexpected SEA or SError on a read access by a load instruction

15-Feb-2023: Changes in document version v4.0

No new or updated errata in this document version.

16-Dec-2022: Changes in document version v3.0

ID	Status	Area	Category	Summary
2740089	New	Programmer	Category B	The core might deadlock during powerdown sequence
2749117	New	Programmer	Category B	Unexpected Dirty and Access flag update when SPE is operated in Discard mode
2763018	New	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation
2778195	New	Programmer	Category B	The core might deadlock during powerdown sequence
2816013	New	Programmer	Category B	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data
2726228	New	Programmer	Category C	TRBE buffer write translation out of context may have incorrect memory attributes
2736657	New	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFI or WFE state
2755356	New	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP
2759988	New	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions
2767982	New	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled
2777198	New	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM
2798494	New	Programmer	Category C	ECC errors in MTE allocation tags may lead to silent data corruption in tag values

29-Jul-2022: Changes in document version v2.0

ID	Status	Area	Category	Summary
2302507	New	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
2450033	Updated	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag, BTB, and L2SMS RAMs
2620954	New	Programmer	Category B	WFI/WFIT instructions can cause speculative MSR write to occur
2631888	New	Programmer	Category B	L1 hardware prefetcher might cause deadlock
2646977	New	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock
2701112	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
2459303	Updated	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails
2590229	New	Programmer	Category C	PCSample field of the Program Counter Sample Register (PMPCSR) might be incorrect
2612737	New	Programmer	Category C	Read to dump the instruction cache contents while in Debug state results in deadlock
2624160	New	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect
2659175	New	Programmer	Category C	WFIT/WFET instructions incorrectly trap after local timeout event has occurred
2666742	New	Programmer	Category C	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data
2667804	New	Programmer	Category C	MTE tag check fail seen on first half of a cache-line crossing load does not get reported
2672593	New	Programmer	Category C	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering
2701260	New	Programmer	Category C	Missing external RAS identification registers
2706749	New	Programmer	Category C	ID_AA64DFR0_EL1.MTPMU reports 0x0 but should report 0xF

08-Apr-2022: Changes in document version v1.0

ID	Status	Area	Category	Summary
2450033	New	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag, BTB, and L2SMS RAMs
2223345	New	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect
2223346	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events
2459303	New	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails

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
2302507	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	r0p0	r0p1
2450033	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag, BTB, and L2SMS RAMs	r0p0	r0p1
2620954	Programmer	Category B	WFI/WFIT instructions can cause speculative MSR write to occur	r0p0	r0p1
2631888	Programmer	Category B	L1 hardware prefetcher might cause deadlock	r0p0	r0p1
2646977	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock	r0p0	r0p1
2701112	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	r0p0	r0p1
2740089	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r0p1	Open
2749117	Programmer	Category B	Unexpected Dirty and Access flag update when SPE is operated in Discard mode	r0p0, r0p1	Open
2763018	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation	r0p0, r0p1	Open
2778195	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r0p1	Open
2816013	Programmer	Category B	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data	r0p0, r0p1	Open
2897503	Programmer	Category B	A load that passes a 32-byte store instruction in MTE precise mode which crosses a page boundary might violate memory ordering	Open, r0p0, r0p1	Open
2908612	Programmer	Category B	Incorrect value of PMSELR register resulting in Unknown Value.	r0p0, r0p1	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
2923985	Programmer	Category B	Branch prediction history not suppressed when switching from low to high EL	r0p0, r0p1	Open
2957258	Programmer	Category B	Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1	r0p0, r0p1	Open
2959655	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly	r0p0, r0p1	Open
3022725	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0, r0p1	Open
2919943	Programmer	Category B (rare)	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level	r0p0, r0p1	Open
2223345	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect	r0p0, r0p1	Open
2223346	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0, r0p1	Open
2459303	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails	r0p0	r0p1
2590229	Programmer	Category C	PCSample field of the Program Counter Sample Register (PMPCSR) might be incorrect	r0p0	r0p1
2612737	Programmer	Category C	Read to dump the instruction cache contents while in Debug state results in deadlock	r0p0	r0p1
2624160	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect	r0p0	r0p1
2659175	Programmer	Category C	WFIT/WFET instructions incorrectly trap after local timeout event has occurred	r0p0	r0p1
2666742	Programmer	Category C	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data	r0p0	r0p1
2667804	Programmer	Category C	MTE tag check fail seen on first half of a cache-line crossing load does not get reported	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2672593	Programmer	Category C	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering	r0p0	r0p1
2701260	Programmer	Category C	Missing external RAS identification registers	r0p0, r0p1	Open
2706749	Programmer	Category C	ID_AA64DFR0_EL1.MTPMU reports 0x0 but should report 0xF	r0p0	r0p1
2726228	Programmer	Category C	TRBE buffer write translation out of context may have incorrect memory attributes	r0p0, r0p1	Open
2736657	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFI or WFE state	r0p0, r0p1	Open
2755356	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	r0p0, r0p1	Open
2759988	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions	r0p0, r0p1	Open
2767982	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	r0p0, r0p1	Open
2777198	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	r0p0, r0p1	Open
2794916	Programmer	Category C	DGH instruction doesn't execute correctly	r0p0, r0p1	Open
2798494	Programmer	Category C	ECC errors in MTE allocation tags may lead to silent data corruption in tag values	r0p0, r0p1	Open
2901622	Programmer	Category C	PMU event MEM_ACCESS_CHECKED_WR incorrectly counts aborted or inactive stores in MTE precise mode	r0p0, r0p1	Open
2904903	Programmer	Category C	Incorrect event count for event 0x80c1 (Non-scalable FP element operations speculatively executed) in PMU	r0p0, r0p1	Open
2910960	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts	r0p0, r0p1	Open
2914109	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption	r0p0, r0p1	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
2928370	Programmer	Category C	PE might report an unexpected SEA or SError on a read access by a load instruction	r0p0, r0p1	Open
2982002	Programmer	Category C	SPE latency counters are corrupted under certain conditions	r0p0, r0p1	Open

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

2302507
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. Fixed in r0p1.

Description

A *Processing Element* (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

This erratum occurs under the following 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.

2450033
Entry into the Full Retention power mode might cause corruption on Itag, BTB, and L2SMS RAMs

Status

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

Description

If a core enters in Full Retention power mode, then the chip enable pin (CE) of Itag RAM, BTB RAM, or L2SMS RAM might be set. Physical RAMs don't support such states, so it leads to corruption when the core comes back to normal power mode and tries to reuse the RAM content.

Configurations Affected

This erratum affects all configurations.
This erratum affects implementations where RAM contents might be corrupted if the CE pin is asserted during retention.

Conditions

The erratum occurs if all the following conditions apply:

1. The PE enters the FULL_RET power state.
2. The Itag, BTB, or L2SMS RAMs are placed into a low-power mode during the PE FULL_RET power state.
3. The PE power state transitions back to ON without going through the OFF power state.

Implications

If the conditions are met, the RAM contents of the itag, BTB, and L2SMS RAMs might be corrupted. As a result, the *Processing Element* (PE) might:

- fetch and execute incorrect opcodes as a result of itag corruption
- predict incorrect targets from corrupted BTB rams
- predict incorrect prefetching targets from corrupted L2SMS RAMs

Workaround

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

2620954
WFI/WFIT instructions can cause speculative MSR write to occur

Status

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

Description

A WFI/WFIT instruction that is not trapped and not treated as a NOP can cause the speculative execution of a MSR instruction, resulting in incorrect system state.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under all of the following conditions:

1. A WFI/WFIT instruction is executed and is not trapped and not treated as a NOP.
2. Later, when that WFI/WFIT instruction flushes the pipeline, a younger MSR is issued at that exact same cycle.

Implications

If the above conditions are met, the MSR instruction can incorrectly write system register state.

Workaround

This erratum can be avoided by serializing after all WFI/WFIT instructions. This can be implemented through execution of the following code at EL3 as soon as possible after boot:

```
;; Insert 'Dispatch Stall until MXQ Empty' AFTER 'WFI/WFIT'
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,= 0xD5030020
MSR S3_6_c15_c8_2,x0
LDR x0,= 0xFFFFCFA0
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x40000004003FF
MSR S3_6_c15_c8_1,x0
ISB
```

2631888
L1 hardware prefetcher might cause deadlock

Status

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

Description

Clock gating logic in the L2 cache might cause internal interface signals to remain asserted, leading to unexpected operation of one of the L1 data cache hardware prefetchers.

Configurations Affected

This erratum affects all configurations

Conditions

Hardware prefetching is enabled.

Implications

If the previous condition is met, unexpected operation, including deadlock, might occur.

Workaround

Disable the affected L1 data cache prefetcher by setting CPUACTLR6_EL1[41] to 'b1. Doing so will incur a performance penalty of ~1%.

Contact Arm for an alternate workaround that impacts power.

2646977
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 r0p0. Fixed in r0p1.

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.

2701112
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 r0p0. Fixed in r0p1.

Description

If a core is fetching instruction 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.

2740089
The core might deadlock during powerdown sequence

Status

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

2749117 Unexpected Dirty and Access flag update when SPE is operated in Discard mode

Status

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

Description

When the *Statistical Profiling Extension* SPE is programmed in Discard mode, PMBPTR and PMBLIMITR need not be programmed with legal values. This could result in unexpected Dirty and Access flag updates on corresponding table descriptors.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Statistical profiling is enabled at the appropriate Exception level and Discard mode is enabled by setting PMBLIMITR.FM = 0b10.

Implications

If the above conditions are met, there could be unexpected Dirty and Access flag updates to block or page descriptors corresponding to the region specified by the PMBPTR and PMBLIMITR.

Workaround

This issue can be mitigated by programming the PMBPTR and PMLIMITR to point to dummy pages that would not trigger translation faults.

2763018 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 r0p0 and r0p1. 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.

2778195
The core might deadlock during powerdown sequence

Status

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

Description

While powering down the *Processing Element* (PE), a CPP instruction without a DSB just before the powerdown sequence might deadlock the PE

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The PE executes a CPP RCTX instruction but does not follow it with a DSB.
2. The PE executes the sequence for powerdown as specified in the *Technical Reference Manual* (TRM).

Implications

If the above conditions are met, the PE might deadlock during the powerdown sequence.

Workaround

Add a DSB instruction following the CPP RCTX instruction. The DSB is necessary to ensure completion of the CPP.

2816013
Execution of STG instructions in close proximity might cause loss of MTE allocation tag data

Status

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

Description

Under certain rare microarchitectural conditions, two or more STG instructions that access the same cacheline but different 32-bytes might not write the MTE allocation tag to memory.

Configurations Affected

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

Conditions

1. Memory tagging is enabled.
2. Two or more STG instructions are executed in close proximity to the same cache line.
3. The STG instructions accesses different 32-bytes locations.

Implications

If the previous conditions are met, then under specific microarchitectural conditions, one of the STG instructions might not write the MTE allocation tag to memory.

Workaround

This erratum can be avoided by setting CPUACTLR5_EL1[14] to 1.

2897503
A load that passes a 32-byte store instruction in MTE precise mode which crosses a page boundary might violate memory ordering

Status

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

Description

Under certain conditions, executing a 32-byte store instruction in MTE precise mode which crosses a page boundary, followed by a load that accesses one or more bytes written by the store, might result in a memory ordering violation.

Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

Conditions

This erratum occurs under the following conditions:

1. A PE executes a 32-byte store in MTE precise mode.
2. The store crosses a page-boundary such that the first page is marked MTE Tagged but the second page is not.
3. A younger load that accesses one or more bytes written by the store to the first page passes the store.
4. A branch mispredict flush occurs around the same time the store completes.

Implications

If the above conditions are met, then under certain microarchitectural conditions, the load might either return stale data or not return data, leading to a deadlock.

Workaround

This erratum can be avoided by setting CPUACTLR4_EL1[8].

Note: Setting CPUACTLR4_EL1[8] to 1 is expected to result in performance degradation for workloads that use MTE when using MTE precise (sync) mode. There is no impact on MTE async & asymmetric modes, or when MTE is disabled.

2908612
Incorrect value of PMSELR_EL0 register resulting in unknown value.

Status

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

Description

PMSELR_EL0 register has an SEL field which value indicates the PMEVCNTR<n>_EL0 event counter that is accessed. The SEL field contains the value <n> that identifies which event counter can be accessed. Under certain conditions PMSELR_EL0 indicates an incorrect value.

Configurations affected

This erratum affects all configurations.

Conditions

1. Write the MDCR_EL2.HPMN register with 0b11110 value.
2. Read the PMCR.N register to check if value is 0b11110.
3. Write the PMCR.E register with 0x1 value.
4. Write PMCNTENSET[n] == 1 (n is a non-zero value, lower than 0b11110).
5. Write the PMSELR_EL0 register with the value n chosen in previous step.
6. Read back the PMSELR_EL0 register.

Implications

If the previous conditions are met, then the SEL field in PMSELR_EL0 register contains an unknown value. The HPMN field in MDCR_EL2 register indicates the number of event counters that are available and can be accessed. We have a masking logic that will provide 0 if the selected event counter is higher than the HPMN value. But for the 0b11110 value, because of an incorrect implementation, it ends up generating an unknown value.

Workaround

This erratum has no workaround.

2923985

Branch prediction history not suppressed when switching from low to high EL

Status

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

Description

Branch prediction history from an attacker in lower *Exception Level* (EL) is not properly suppressed when switching to a victim at higher EL. This causes the victim to unexpectedly speculate to a section of its own code that contains instructions that cause a side effect (such as a cache miss) which is later observable by the attacker.

Configurations affected

This erratum affects all configurations.

Conditions

When switching from lower EL to higher EL and the following CPUACTLR4 bits are configured as follows:

- Bit 11: BHB_SUPPRESS_AT_VEC_RESTART_DIS is set to 0.
- Bit 10: BHB_FLUSH_AT_VEC_RESTART_EN is set to 0.

Implications

An attacker running at lower EL might affect the behavior of a victim at higher EL, causing the victim to incorrectly (unexpectedly) speculate to one of its targets, which in turn can cause a side effect (such as a cache miss) observable by the attacker. A carefully crafted attack might result in confidential or sensitive information being leaked by the victim.

Workaround

The recommended hardware workaround is to disable BHB suppress, and to enable BHB flush. This can be done via CPUACTLR4 as follows:

- Set bit 11: BHB_SUPPRESS_AT_VEC_RESTART_DIS to 1.
- Set bit 10: BHB_FLUSH_AT_VEC_RESTART_EN to 1.

Using the above combination, the history register will be cleared on low to high EL transitions, precluding the attack, but with a negligible performance impact.

2957258
Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1

Status

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

Description

In EL2/EL3, reads of MPIDR_EL1 and MIDR_EL1 might incorrectly virtualize which register to return when reading the value of MPIDR_EL1/VMPIDR_EL2 and MIDR_EL1/VPIDR_EL2, respectively.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. An exception entry to EL2 or EL3 occurs
2. No context synchronizing event (such as an ISB) has occurred since the last exception entry
3. An MRS instruction is executed to read either MPIDR_EL1 or MIDR_EL1

Implications

If the previous conditions are met, then the core might not correctly choose what it should return:

- when executing a read to MPIDR_EL1, it might return either MPIDR_EL1 (correctly) or VMPIDR_EL2 (incorrectly)
- when executing a read to MIDR_EL1, it might return either MIDR_EL1 (correctly) or VPIDR_EL2 (incorrectly)

Workaround

This erratum can be avoided by inserting an ISB prior to an MRS read to either MPIDR_EL1 and MIDR_EL1. Performance impact is expected to be negligible in real systems. This sequence can be implemented through execution of the following code at EL3 as soon as possible after boot:

```
    // add ISB before MRS reads of MPIDR_EL1/MIDR_EL1
    LDR x0,=0x1
    MSR S3_6_c15_c8_0,x0 // MSR CPUPSELR_EL3, X0
    LDR x0,=0xd5380000
    MSR S3_6_c15_c8_2,x0 // MSR CPUPOR_EL3, X0
    LDR x0,=0xFFFFFF40
    MSR S3_6_c15_c8_3,x0 // MSR CPUPMR_EL3, X0
    LDR x0,=0x000080010033f
    MSR S3_6_c15_c8_1,x0 // MSR CPUPCR_EL3, X0
    ISB
```

2959655
PE executing DRPS during Debug Halt under Double Fault condition will not execute properly

Status

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

Description

When a DRPS instruction is executed in Debug Halt state, a double fault should cause implicit ESB according to the *Arm Architecture Reference Manual for A-profile architecture* when (SCR_EL3.EA == '1' && SCR_EL3.NMEA == '1' && PSTATE.EL == **EL3**). However, the *Processing Element* (PE) will only execute part of the instruction for this case.

Configurations affected

This erratum affects all configurations with double fault extension.

Conditions

This erratum occurs under the following conditions:

1. The PE is in Debug Halt state.
2. Software is currently executing at EL3 Exception level.
3. SCTLR_EL3.IESB == '0'
4. SCR_EL3.EA == '1' && SCR_EL3.NMEA == '1' indicating double fault.

Implications

The DRPS instruction is not executed correctly.

Workaround

When executing a DRPS instruction in EL3, set SCTLR_EL3.IESB to override double fault. Doing this will force the correct DRPS execution sequence to occur.

3022725
SPE might write to pages which lack write permission at Stage-1 or Stage-2

Status

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

Description

The *Statistical Profiling Extension* (SPE) uses the Stage-1 translation regime of the owning exception level in the owning Security state. Due to this erratum, the SPE might write to memory which lacks write permission at Stage-1 and/or Stage-2 of the owning exception level's translation regime, without raising a fault.

Configurations affected

This erratum affects all configurations that support SPE.

Conditions

This erratum occurs under the following conditions:

1. The SPE buffer is enabled.
2. Registers PMBPTR_EL1 and PMBLIMITR_EL1 are configured to include a virtual address VA_X.
3. A valid Stage-1 translation exists for the virtual address VA_X.
4. If Stage-2 is enabled, a valid Stage-2 translation exists for the intermediate physical address IPA_X for the virtual address VA_X.
5. At least one of the following conditions is true:
   a. The Stage-1 translation for VA_X lacks write permission.
   b. The Stage-2 translation for IPA_X lacks write permission.
6. None of the following apply:
   a. Stage-1 hardware dirty bit management is enabled.
   b. Stage-2 is enabled, and Stage-2 hardware dirty bit management is enabled.

Implications

The SPE might write to VA_X rather than generating a fault. This might allow malicious software with control over SPE to corrupt memory for which it is not intended to have write access to.

Workaround

No hardware workaround is available.

A hypervisor at EL2 should not give virtual machines control of SPE unless the hypervisor can handle writes to any pages mapped at Stage-2.

An OS kernel at EL1 or EL2 should not configure the SPE buffer to contain any page which might lack write permission at Stage-1.

No current software is expected to have this problem.

Category B (rare)

2919943
PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level

Status

Fault Type: Programmer Category B (Rare)
Fault Status: Present in r0p0 and r0p1. Open.

Description

Under certain conditions, the *Processing Element* (PE) might incorrectly detect a Watchpoint debug event instead of a Data Abort exception when a memory access spans multiple pages. The Data Abort is detected for the first page and the Watchpoint debug event is associated with the second page. The Watchpoint debug event detection might route the Data Abort to the incorrect target Exception level or cause the PE to enter Debug state.

Note the contents of the ESR and FAR registers capture the information associated with the Data Abort.

Configurations affected

This erratum affects all configurations.

Conditions

1. Watchpoints are enabled.
2. The PE executes a page split access that generates a Data Abort on the first page and a Watchpoint match on the second page.
3. The PE executes a younger load instruction that generates an external abort which coincides with a 1 cycle window when processing the Data Abort and Wathchpoint debug event.

Implications

If the previous conditions are met and EDSCR.HDE is set (enables Halting Debug on Watchpoint debug event), then the PE will enter Debug state rather than taking a Data Abort exception.

If EDSCR.HDE is not set, the PE might route the abort to the incorrect Exception level:

- If MDCR_EL2.TDE == 0, a stage 2 Data Abort might result in a Data Abort exception taken erroneously to EL1.

    - The rarity of PE internal timings required to exhibit this bug is comparable to *Reliability, Availability, and Serviceability* (RAS) error FIT rates. Expected outcome is a kernel panic that will kill the process.
- If MDCR_EL2.TDE == 1, a stage 1 Data Abort might result in a Data Abort exception taken erroneously to EL2.
    - This scenario is containable within a hypervisor via the software workaround outlined below.

Workaround

There is no complete workaround for this erratum. A partial software workaround addresses the more serious scenario of a stage 1 Data Abort resulting in a Data Abort exception taken erroneously to EL2 without updating HPFAR_EL2.

EL2 can protect against this case as follows:

- Reserve one bit of IPA space so that VTCR_EL2.PS is never the maximum supported.
- Write all 1's to HPFAR_EL2[63:0] before entering EL1 or EL0.
- Exceptions to EL2 due to this erratum that should have set HPFAR_EL2 will instead use an out of range IPA. The guest should be restarted as the conditions for this erratum are rare and are not likely to be encountered again.

Category C

2223345
MPAM value associated with instruction fetch might be incorrect

Status

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

Description

Under some scenarios, the MPAM value associated with an instruction fetch request might be incorrect when context changes.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An Instruction fetch request is attempted before a context switch but is not completed until after a context switch.

Implications

The MPAM value associated with the instruction fetch request might be incorrect.

Workaround

There is no workaround.

2223346
Noncompliance with prioritization of Exception Catch debug events

Status

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

2459303 Checked stores in precise mode might fail to report tag check fails

Status

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

Description

When *Memory Tagging Extension* (MTE) is used in Synchronous mode, the Tag-Check-read performed by a store instruction might fail to be ordered with respect to older instructions with acquire semantics or a DMB (or DSB). This ordering violation might lead to the store instruction writing to memory in cases where it should have failed its tag check.

Configurations Affected

This erratum affects configurations with BROADCASTMTE=1.

Conditions

The erratum occurs when all the following conditions are met:

1. MTE is enabled in precise checking mode (SCTLR_ELx.TCF='b01).
2. Allocation tags read by the *Processing Element* (PE) are modified by another PE.
3. An instruction RW1 is executed and either
   - R1 is generated by an instruction with Acquire semantics or appears in program order before a DMB LD (or DSB), or
   - RW1 appears in program order before a DMB FULL (or DSB)
4. A tag-checked store instruction W3, performing a Tag-Check-read R2, is executed and either
   - W3 appears in program order after R1 with acquire semantics, or
   - W3 appears in program order after DMB LD/FULL (or DSB)

Implications

When the above conditions are met, the Tag-Check-read R2 performed by W3 might be observed before R1.

- If the observed tag value does not cause a Tag Check Fault, W3 will update memory even in cases where observation in the correct order should have resulted in a tag Check Fault.
- If the observed tag value causes a Tag Check Fault, there are no implications and the PE behaves as expected.

This will lead to a very small percentage of escapes in the tag checking logic, sometimes causing a tag check pass when it should be a tag check fail.

Workaround

There is no workaround.

2590229
PCSample field of the Program Counter Sample Register (PMPCSR) might be incorrect

Status

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

Description

PCSample field in PMPCSR register indicates the sampled instruction address value. It indicates the target address each time the processing element (PE) executes an instruction that can be currently sampled or which retires. Under certain conditions, PMPCSR returns an incorrect value for the PCSample field.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The PE is not in Debug state.
2. PCSample based profiling is not prohibited.
3. PMPCSR[31:0] is at least read once since PE left reset state.
4. Two branch instructions are executed in close program order proximity.
5. Debugger reads the PMPCSR[31:0].

Implications

If the above conditions are met then the PCSample field of the PMPCSR register contains the target address of the older branch instruction instead of the target address of the younger branch instruction.

Workaround

There is no workaround.

2612737
Read to dump the instruction cache contents while in Debug state results in deadlock

Status

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

Description

In Debug state, an access to read the instruction cache data contents using SYS_IMP_RAMINDEX will not complete and will deadlock any ITR transactions that follow.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs if all the following conditions apply:

1. The PE enters Debug state.
2. User sets SYS_IMP_RAMINDEX RAM_ID field to 0x1 in order to select the read of instruction cache contents, and performs the read.

Implications

The instruction cache read deadlocks, and the debugger might lose control.

Workaround

This erratum can be avoided by the debugger if the instruction cache is not read when the core is in Debug state.

2624160
FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect

Status

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

Description

A *Scalable Vector Extension* (SVE) first fault contiguous load instruction that encounters a Tag Check fail when accessing the first active element and a watchpoint match on one of the non-first active elements can generate a Data abort exception with incorrect value in FAR_ELx.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging and watchpoints are enabled.
2. An SVE first fault contiguous load instruction accesses memory and generates a Data Abort exception due to Tag Check fail on the first active element.
3. There is a watchpoint match on one of the non-first active elements.

Implications

If the above conditions are met, a Data Abort exception will be generated with an incorrect value in FAR_ELx. ESR_ELx will indicate Synchronous Tag Check Fault.

Workaround

This erratum has no workaround.

2659175 WFIT/WFET instructions incorrectly trap after local timeout event has occurred

Status

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

Description

WFIT and WFET instructions do not have correct prioritization of first checking whether a local timeout event has occurred prior to checking if a trap should occur. Instead of being treated as a NOP after a local timeout event has occurred, these instructions can cause a trap exception to occur.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under all of the following conditions:

1. A local timeout event occurs.
2. A WFIT/WFET is executed and it is trapped.

Implications

If the above conditions are met, a trap occurs erroneously when the WFIT/WFET should behave as a NOP. Software is expected to already handle WFIT/WFET traps under normal race conditions.

Workaround

There is no workaround.

2666742 Execution of STG instructions in close proximity might cause loss of MTE allocation tag data

Status

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

Description

Under certain rare micro-architectural conditions, two or more STG instructions that access the same cacheline but different 32-bytes might not write the MTE allocation tag to memory in the presence of an ECC error to the same cache index.

Configurations Affected

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

Conditions

1. Memory tagging is enabled.
2. Two or more STG instructions are executed in close proximity to the same cache line
3. The STG instructions accesses different 32-bytes locations.
4. An L2 fill for a different cacheline but to the same index has a single bit data error that could have otherwise caused a capacity evict of the cacheline accessed by the STG instructions

Implications

If the above conditions are met, then under specific micro-architectural conditions, the MTE allocation tag might not be written to memory, resulting in a silent corruption of the MTE tag

Workaround

If desired, this erratum can be avoided by setting CPUACTLR5_EL1[13] to 1.

Note: setting CPUACTLR5_EL1[13] to 1 is expected to result in a small performance degradation for workloads that use MTE (approximately 1.6% when using MTE imprecise mode, 0.9% for MTE precise mode).

2667804 MTE tag check fail seen on first half of a cache-line crossing load does not get reported

Status

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

Description

Under some unusual microarchitectural conditions, tag check fail seen on first half of a cache-line crossing load does not get reported.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging is enabled
2. Cache-line crossing load is executed that fails tag check on first half of the access
3. Unusual microarchitectural conditions occur

Implications

If the above conditions are met, precise checked loads that see tag mismatch will not report an exception and imprecise checked loads will not update the TFSR register.

Workaround

This erratum has no workaround.

2672593 MTE checked load might read an old value of allocation tag by not complying with address dependency ordering

Status

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

Description

Under some unusual micro-architectural conditions, checked load might read an old value of allocation tag by not complying with address dependency ordering.

Configurations Affected

All configurations are affected.

Conditions

The erratum occurs when all the following apply:

1. Initially, memory location M has allocation tag A.
2. *Processing Element* x (PEx) stores to M using allocation tag A.
3. PEy changes the allocation tag of M from A to B.
4. PEx makes a checked load from M using allocation tag A, with a dependency such that it should observe allocation tag B.

Implications

If the above conditions are met, PEx may not observe the new allocation tag for the memory location and may fail to report a tag check fail.

Workaround

This erratum has no workaround.

2701260
Missing external RAS identification registers

Status

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

Description

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when accessing one of the following registers through the memory-mapped interface to the RAS registers:

- ERRGSR
- ERRDEVAFF
- ERRDEVARCH
- ERRDEVID

Implications

When the above conditions are met, reads of the affected registers always return 0.

Workaround

This erratum has no workaround.

2706749
ID_AA64DFR0_EL1.MTPMU reports 0x0 but should report 0xF

Status

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

Description

Executing a MRS instruction to read the field MTPMU in the system register ID_AA64DFR0_EL1 returns 0x0 rather than the expected value of 0xF (since FEAT_PMUv3 is supported and FEAT_MTPMU is not supported).

NOTE: PMEVTYPER<n>_EL0.MT is appropriately implemented as RES0.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under all of the following conditions:

1. A MRS instruction is executed to system register ID_AA64DFR0_EL1 that does not cause an exception

Implications

If the above conditions are met, the MRS instruction returns incorrect read data for the field MTPMU in ID_AA64DFR0_EL1.

Workaround

This erratum has no workaround.

2726228
TRBE buffer write translation out of context may have incorrect memory attributes

Status

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

Description

When TRBLIMITR_EL1.nVM = 1, TBE_OWNING_EL = EL1, and TRBE requests a translation while the *Processing Element* (PE) is executing in EL2 or EL3, and cache is disabled by HCR_EL2.CD = 1, memory attribute may not be Non-cacheable.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. TRBLIMITR_EL1.nVM is set to 1.
2. MDCR_EL2.E2TB is set to 0b10 or 0b11.
3. HCR_EL2.CD is set to 1.
4. The PE is executing in EL2 or EL3.
5. TRBE requests a translation for a buffer write.

Implications

Memory attributes for any write access by TRBE to that translation may not be forced to Non-cacheable.

Workaround

Use of HCR_EL2.CD is not expected to be common. If a workaround is needed, do not allow TRBE to be given to a VM machine.

2736657
AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFI or WFE state

Status

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

Description

The core frequency cycles Activity Monitor Unit (AMU) event may not count correctly when the core is in Wait For Interrupt (WFI) or Wait For Event (WFE) state and the clocks in the core are enabled.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The architected activity monitor counter register 0 (AMEVCNTR00) is enabled.
2. The core executes WFI or WFE instructions.
3. The clocks in the core are never disabled, or
4. The clocks in the core are temporarily enabled without causing the core to exit WFI or WFE state due to one of the following events:
    - A system snoop request that must be serviced by the core L1 data cache or the L2 cache.
    - A cache or Translation Lookaside Buffer (TLB) maintenance operation that must be serviced by the core L1 instruction cache, L1 data cache, L2 cache, or TLB.
    - An access on the Utility bus interface.
    - A Generic Interrupt Controller (GIC) CPU access or debug access through the Advanced Peripheral Bus (APB) interface.

Implications

The core frequency cycles AMU event will continue to increment when clocks are enabled even though the core is in WFI or WFE state. Arm expects this to be a minor issue as the resulting discrepancies will likely be negligible from the point of view of consuming these counts in the system firmware at the 1ms level.

Workaround

There is no workaround.

2755356
Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r0p1. 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.

2759988
Incorrect decoding of SVE version of PRF* scalar plus scalar instructions

Status

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

Description

Scalar plus Scalar and Scalar plus Immediate forms of the *Scalable Vector Extension* (SVE) PRF may not prefetch from the correct address.

- PRFB (scalar plus scalar)
- PRFH (scalar plus scalar)
- PRFW (scalar plus scalar)
- PRFD (scalar plus scalar)
- PRFB (scalar plus immediate)
- PRFH (scalar plus immediate)
- PRFW (scalar plus immediate)
- PRFD (scalar plus immediate)

Configurations Affected

This erratum affects all configurations.

Conditions

1. Any of the above instructions are executed without trapping

Implications

All affected instructions are software prefetches which do not affect architectural state in any way (including suppression of any translation faults). Thus this erratum will not affect the functional operation of the CPU.

Workaround

No workaround is expected to be necessary.

2767982
Incorrect timestamp value reported in SPE records when timestamp capture is enabled

Status

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

Description

The timestamp value that is captured in the *Statistical Profiling Extension* (SPE) records may be incorrect.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Timestamp capture is enabled for SPE records at the appropriate Exception level by setting PMSCR_EL1.TS or PMSCR_EL2.TS.

Implications

If the above conditions are met, then the timestamp value reported in the SPE records might be stale (off by one tick) or zero in some cases.

Workaround

There is no workaround.

2777198
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 and r0p1. 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.

2794916
DGH instruction doesn't execute correctly

Status

Fault Type; Programmer Category C
Fault Status: Present in r0p0 and r0p1. Open

Description

DGH instructions are executed as PSBs. The DGH target address is ignored.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. A DGH instruction is executed.

Implications

If Profiling is not enabled, then the PSB will execute as a NOP. Performant code sequences that depend on DGH for explicit memory management will not see the expected speedup, but will see no additional slowdown.

If Profiling is enabled, then the PSB instruction might take up to tens of cycles to complete, causing an additional slowdown.

Since neither DGH nor PSB affect the architected state, there is no functional problem.

Workaround

No workaround is expected to be necessary.

2798494 ECC errors in MTE allocation tags might lead to silent data corruption in tag values

Status

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

Description

Streaming writes that require *Memory Tagging Extension* (MTE) tags for tag checking or merging with data receive allocations tags that are flagged as poisoned might lead to the *Processing Element* (PE) caching data and tags with no indication that the tags are poisoned. This might lead to silent data corruption on the allocation tags.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The PE performs a streaming write (a write of 64 contiguous bytes gathered from multiple store or DC ZVA operations).
2. Streaming write requires MTE tag check or hits in the PE caches to a line that contains MTE allocation tags.
3. MTE allocations tags contain an indication of an error (uncorrectable ECC error or poison flag).

Implications

If the above conditions are met, the PE might merge the streaming write data and the MTE allocation tags containing an error and write data and allocation tags to a cache without marking the tags as poisoned. This can lead to silent data corruption to future consumers of the MTE allocation tags, which might result in incorrect MTE tag check results. The net effect is an increase in the SDC FIT rate of the PE.

There is still substantial benefit being gained from the ECC logic.

Workaround

There is no workaround.

2901622 PMU event MEM_ACCESS_CHECKED_WR incorrectly counts aborted or inactive stores in MTE precise mode

Status

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

Description

The MEM_ACCESS_CHECKED_WR PMU events increment incorrectly when accessing a tagged page, although the write is aborted.

Configurations affected

This erratum affects configurations with BROADCASTMTE = 1.

Conditions

This erratum occurs if the following conditions apply:

1. A store accesses an MTE tagged page in MTE precise mode.
2. The write is either aborted or inactive due to SVE predication.

Implications

If the previous conditions are met, the PMU event might increment inaccurately.

Workaround

This erratum has no workaround.

2904903 Incorrect event count for event 0x80c1 (Non-scalable FP element operations speculatively executed) in PMU

Status

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

Description

On programming the event 80c1 in PMEVTYPER<n>_EL0 register, and when ensured that a non-scalable FP element based operations are speculatively executed; under certain conditions PMEVCNTR<n>_EL0.CNTR indicates the incorrect value.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. PMCR_EL0.E. is 0
2. MDCR_EL2.HPME is 1.
3. MDCR_EL2.HPMN is a value less than the value of PMCR_EL0.N.
4. For some value of n greater than or equal to MDCR_EL1.HPMN and less than PMCR_EL0.N:
    a. PMCNTENSET[n] is 1.
    b. PMEVTYPER<n>_EL0.evtCount is 0x80c1.
5. The Event 0x80c1 is generated. This event counts speculatively executed operations floating point operations generated by floating point or Advanced SIMD instructions.

Implications

The event counter gives an incorrect value for the programmed event on PMEVCNTR<n>_EL0.CNTR.

Workaround

There is no workaround.

2910960 L2D_CACHE_WB_CLEAN overcounts

Status

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

Description

Counting of the L2D_CACHE_WB_CLEAN event includes transfer of data directly to another *Processing Element* (PE) using the AMBA CHI Direct Cache Transfer mechanism.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The PE processes a forwarding snoop from the DSU or *Fully coherent Home Node* (HN-F) and sends data directly to another PE using a CompData message.

Implications

If the previous condition is met, the PE will count the L2D_CACHE_WB_CLEAN event contrary to the architectural specification of this event.

Workaround

No workaround is required for this erratum.

2914109
Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption

Status

Fault type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Open.

Description

A PE accessing a same physical memory location with mismatched Shareability attributes and requiring a read of *Memory Tagging Extension* (MTE) tags might result in data corruption.

Configurations affected

This erratum affects all configurations.

Conditions:

This erratum occurs under the following conditions:

1. PE accesses a physical memory location using cacheable and Non-shareable attributes.
2. PE accesses the same physical address using cacheable and shareable attributes with MTE checking enabled.

Implications

If the previous conditions are met, the PE might expose stale data from the PE caches established by a Non-shareable access. This data might become visible to shareable observers in the same Shareability domain, even if the PE performs the required cache maintenance for ensuring ordering and coherency when aliasing Shareability.

Workaround

Arm expects that operating systems do not use mismatched Shareability attributes for aliases of the same memory location for tagged pages.

2928370 PE might report an unexpected SEA or SError on a read access by a load instruction

Status

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

Description

Under certain micro-architectural conditions, a load executing on a *Processing Element* (PE) might incorrectly consume data poison or DErr/NDErr that was meant for an instruction fetch or descriptor fetch for an unrelated translation table walk.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs if the following conditions apply:

1. The PE executes a load instruction.
2. An instruction fetch or descriptor fetch for an unrelated translation table walk returns poisoned data or generates a DErr/NDErr.

Implications

If the previous conditions are met, then the PE might incorrectly signal SEA or SError on the load instruction, but the data returned by the load will be correct.

Workaround

There is no workaround for this erratum.

2982002
SPE latency counters are corrupted under certain conditions

Status

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

Description

Under certain conditions, the dispatch to issue and dispatch to completion latency counters for certain Statistical Profiling samples might be corrupted.

Configurations affected

This erratum affects all configurations.

Conditions

1. Statistical profiling is enabled at the appropriate Exception level.
2. The first instruction sampled is one of the following instructions:
    - FADDA
    - BFMMLA
    - FDIV
    - FSQRT
3. The sample gets flushed under certain micro-architectural conditions.
4. The next sample of one of the above instructions might capture incorrect latency values.

Implications

If the above conditions are met, the dispatch to issue and dispatch to completion counts for certain samples of FADDA, BFMMLA, FDIV, or FSQRT in the *Statistical Profiling Extension* (SPE) buffer might be corrupted.

Workaround

There is no workaround.