Arm® Neoverse V3AE Core (MP172)

Software Developer Errata Notice

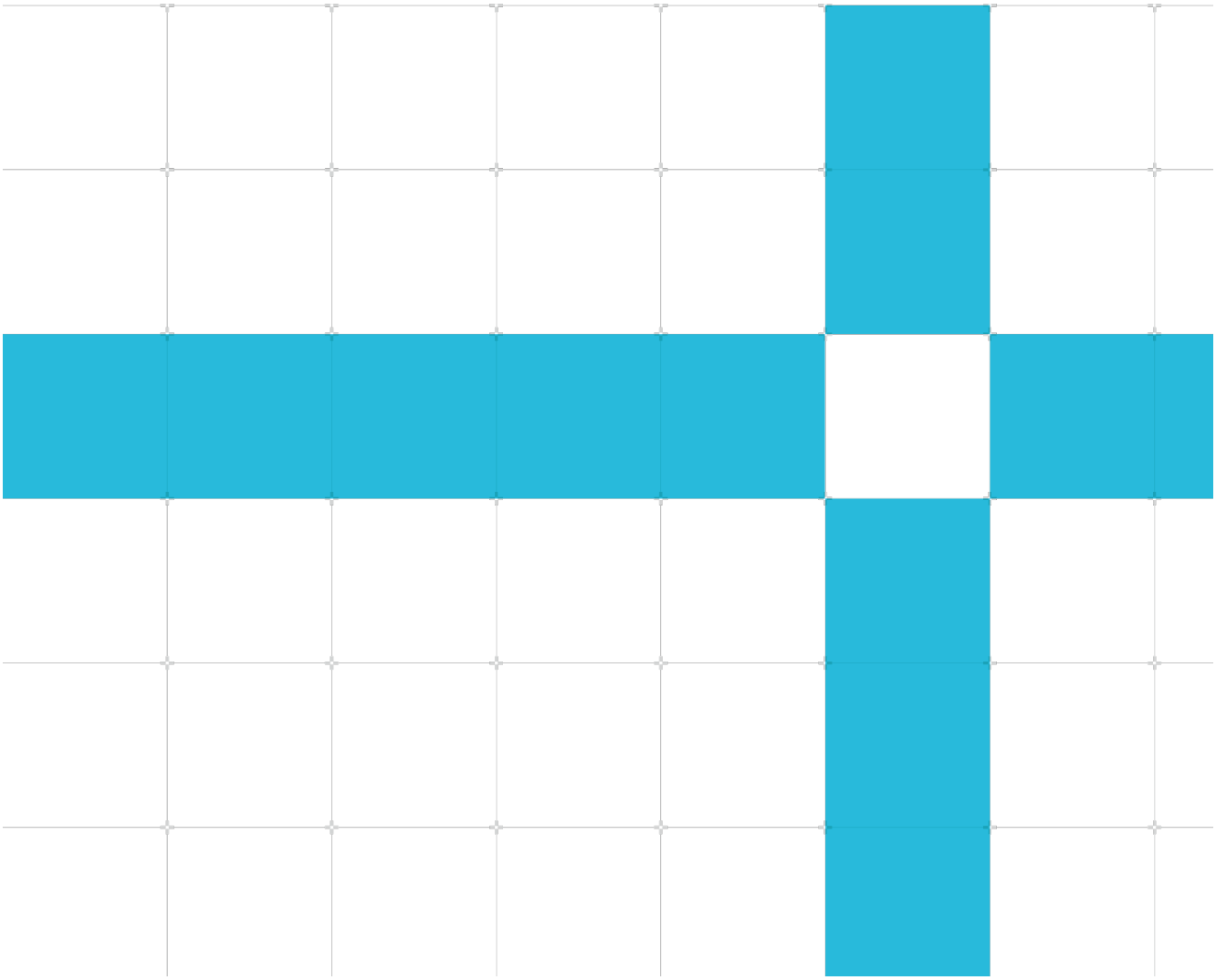
Date of issue: March 13, 2024

Non-Confidential

Document version: 6.0

Document ID: SDEN-2615521

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



Non-confidential proprietary notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality status

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

Unrestricted Access is an Arm internal classification.

Product status

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® Neoverse V3AE Core (MP172) , 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.

March 13, 2024: Changes in document version v6.0

No new or updated errata in this document version.

February 21, 2024: Changes in document version v5.0

ID	Status	Area	Category	Summary
3157034	New	Programmer	Category B	Deadlock in FULL_RET power mode if core power domain boundary is clamped
3177202	New	Programmer	Category C	BROADCASTMTE CPU Boot-time pin does not cause DC CIGDPAPA to correctly UNDEF

November 01, 2023: Changes in document version v4.0

ID	Status	Area	Category	Summary
2930980	Updated	Programmer	Category B	Direct write to ACCDATA_EL1 only observable after a context synchronizing event
2970647	Updated	Programmer	Category B	Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1
2982188	Updated	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly
2982000	Updated	Programmer	Category B	Branch prediction history not suppressed when switching from low to high EL
3030120	Updated	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
3053180	Updated	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock
3090385	New	Programmer	Category B	The CPU could deadlock under certain micro-architectural conditions
3097812	New	Programmer	Category B	Power off transition might deadlock if FULL_RET is enabled
2986656	Updated	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
2933585	Updated	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts
2940264	Updated	Programmer	Category C	PMU event MEM_ACCESS_CHECKED_WR incorrectly counts aborted or inactive stores in MTE precise mode
2940266	Updated	Programmer	Category C	PE might report an unexpected SEA or SError on a read access by a load instruction
2963918	Updated	Programmer	Category C	Incorrect event count for event 0x80c1 (Non-scalable FP element operations speculatively executed) in PMU
2982003	Updated	Programmer	Category C	SPE latency counters are corrupted under certain conditions
3071658	New	Programmer	Category C	TagMatch responses with error indication do not generate a SError abort

September 11, 2023: Changes in document version v3.0

ID	Status	Area	Category	Summary
2930980	New	Programmer	Category B	Direct write to ACCDATA_EL1 only observable after a context synchronizing event
3030120	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
3053180	New	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock

July 14, 2023: Changes in document version v2.0

ID	Status	Area	Category	Summary
2970647	New	Programmer	Category B	Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1
2982188	New	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly
2982000	New	Programmer	Category B	Branch prediction history not suppressed when switching from low to high EL
2986656	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
2921482	New	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption
2933585	New	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts
2940264	New	Programmer	Category C	PMU event MEM_ACCESS_CHECKED_WR incorrectly counts aborted or inactive stores in MTE precise mode
2940266	New	Programmer	Category C	PE might report an unexpected SEA or SError on a read access by a load instruction
2963918	New	Programmer	Category C	Incorrect event count for event 0x80c1 (Non-scalable FP element operations speculatively executed) in PMU
2982003	New	Programmer	Category C	SPE latency counters are corrupted under certain conditions

March 27, 2023: Changes in document version v1.0

ID	Status	Area	Category	Summary
2928513	New	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect
2936120	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events

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
2930980	Programmer	Category B	Direct write to ACCDATA_EL1 only observable after a context synchronizing event	r0p0	r0p1
2970647	Programmer	Category B	Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1	r0p0	r0p1
2982000	Programmer	Category B	Branch prediction history not suppressed when switching from low to high EL	r0p0	r0p1
2982188	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly	r0p0	r0p1
3030120	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0	r0p1
3053180	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	r0p0	r0p1
3090385	Programmer	Category B	The CPU could deadlock under certain micro-architectural conditions	r0p0	r0p1
3097812	Programmer	Category B	Power off transition might deadlock if FULL_RET is enabled	r0p0	r0p1
3157034	Programmer	Category B	Deadlock in FULL_RET power mode if core power domain boundary is clamped	r0p1	Open
2986656	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
2921482	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption	r0p0	Open
2928513	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect	r0p0	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
2933585	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts	r0p0	r0p1
2936120	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0	Open
2940264	Programmer	Category C	PMU event MEM_ACCESS_CHECKED_WR incorrectly counts aborted or inactive stores in MTE precise mode	r0p0	r0p1
2940266	Programmer	Category C	PE might report an unexpected SEA or SError on a read access by a load instruction	r0p0	r0p1
2963918	Programmer	Category C	Incorrect event count for event 0x80c1 (Non-scalable FP element operations speculatively executed) in PMU	r0p0	r0p1
2982003	Programmer	Category C	SPE latency counters are corrupted under certain conditions	r0p0	r0p1
3071658	Programmer	Category C	TagMatch responses with error indication do not generate a SError abort	r0p0	r0p1

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

2930980
Direct write to ACCDATA_EL1 only observable after a context synchronizing event

Status

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

Description

A direct read from ACCDATA_EL1 does not observe the value written by a direct write to ACCDATA_EL1 until after a context synchronizing event.

Configurations affected

This erratum affects configurations with FEAT_LS64 enabled.

Conditions

This erratum occurs if the following conditions apply:

1. A MSR is executed to write a value to ACCDATA_EL1.
2. A MRS is executed to read from ACCDATA_EL1 and no context synchronizing event has occurred since the last write.

Implications

If the previous conditions are met, the read will not see the last value written to ACCDATA_EL1.

Workaround

This erratum can be avoided by inserting an ISB prior to a MRS read to ACCDATA_EL1.

2970647
Incorrect virtualization of reads to MPIDR_EL1 and MIDR_EL1

Status

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

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
```

2982000
Branch prediction history not suppressed when switching from low to high EL

Status

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

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.

2982188
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. Fixed in r0p1.

Description

Whenever there is a *Debug Restore Processor State* (DRPS) executed in Debug Halt state, a double fault should cause implicit *Error Synchronization Barrier* (ESB) per the 'Arm® Architecture Reference Manual for A-profile architecture' when (SCR_EL3.EA == '1' && SCR_EL3.NMEA == '1' && PSTATE.EL == EL3). However, the 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. Debug Halt state
2. Currently in EL3 exception level
3. SCTLR_EL3.IESB == '0'
4. SCR_EL3.EA == '1' && SCR_EL3.NMEA == '1' indicating double fault

Implications

Execution of DRPS will execute partial IESB operation without DRPS operation.

Workaround

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

3030120
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. Fixed in r0p1.

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.

3053180
Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock

Status

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

Description

Under certain conditions, changing block size without break-before-make or mis-programming the contiguous bit can lead to an interruptible livelock in violation of FEAT_BBM level 2 requirements until TLB maintenance is performed.

Configurations affected

This erratum affects all configurations.

Conditions

1. The contiguous bit is mis-programmed for a set of contiguous Stage-1 or Stage-2 translation table entries.
2. A load or store crosses a page boundary within a contiguous address range such that an access for one page is translated by a translation table entry with the contiguous bit set and an access for another page is translated via a translation table entry with the contiguous bit clear.

or

1. A Stage-1 or Stage-2 translation table entry is modified without break-before-make such that a VA or IPA which was previously translated by a Page or Block entry is subsequently translated via a larger Block entry.
2. No TLB maintenance is performed to remove TLB entries for the stale Page or Block entry.
3. A load or store crosses a page boundary such that accesses for either page could be translated via the new block entry, and at least one access could have been translated by a distinct Page or Block entry prior to modification.

Implications

When the previous conditions are met, the load or store instruction will stall indefinitely without raising a fault. During the stall, the load or stall can be interrupted.

Workaround

Where software which manages the translation tables cannot ensure that it is not subject to the stall conditions, or where stalling is unacceptable, software which manages the translation tables should ignore **ID_AA64MMFR2_EL1.BBM** and always follow a break-before-make approach.

Where software which manages the translation tables can ensure that it is not subject to the stall conditions, and it is acceptable to transiently stall lower privileged software, software which manages the translation tables should minimize the period for which the contiguous bit is mis-programmed and minimize the period between modifying a translation table entry and invalidating TLB entries for the previous translation table entry.

3090385
The CPU might deadlock under certain micro-architectural conditions

Status

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

Description

Under certain micro-architectural conditions the *Processing Element* (PE) might deadlock while executing instructions that write PSTATE.{N,Z,C,V} conditional flags in the presence of a precisely timed branch misprediction.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions apply:

- The PE executes many instructions that update the PSTATE.{N,Z,C,V} conditional flags with a latency of 2 cycles.
- The PE executes many instructions that update the PSTATE.{N,Z,C,V} conditional flags with a latency of 1 cycle.
- PSTATE.SSBS = 0 (not strictly needed but significantly increases deadlock potential)
- The PE executes a branch that mispredicts.
- Additional internal issue queue occupancy and timing conditions need to be met.

Implications

If the previous conditions are met, under certain micro-architectural conditions the PE might deadlock.

Workaround

The deadlock can be avoided in all cases at the cost of some performance by setting the following registers, early in the EL3 boot sequence: CPUACTLR3_EL1[14:13]=0b11, CPUACTLR_EL1[52]=1 . Expected performance degradation is < 0.5%, but isolated benchmark components might see higher impact.

3097812
Power off transition might deadlock if FULL_RET is enabled

Status

Fault type: Programmer Category B
Fault status: Present in r0p0. Fixed in r0p1.

Description

If the interrupt/event retention bits are non-zero in the IMP_CPUPWRCTLR_EL1 register to enable the FULL_RET power mode, then the power transition from ON to OFF/OFF_EMU might deadlock.

Configurations affected

All configurations are affected.

Conditions

This erratum occurs under the following conditions:

1. The IMP_CPUPWRCTLR_EL1.WFI_RET_CTLR or IMP_CPUPWRCTLR_EL1.WFE_RET_CTLR fields are non-zero, enabling the FULL_RET power mode.
2. Software sets the IMP_CPUPWRCTLR_EL1.CORE_PWRDN_EN bit to request a powerdown.
3. The core executes a WFI or WFE instruction.
4. The PPU starts a power transition to OFF or OFF_EMU.

Implications

If the erratum condition occurs, then the power transition to OFF or OFF_EMU might deadlock.

Workaround

For workaround as part of the power down sequence, EL3 software should set the IMP_CPUPWRCTLR_EL1.WFI_RET_CTLR and IMP_CPUPWRCTLR_EL1.WFE_RET_CTLR fields to zero before it sets the IMP_CPUPWRCTLR_EL1.CORE_PWRDN_EN field.

3157034
Deadlock in FULL_RET power mode if core power domain boundary is clamped

Status

Fault type: Programmer Category B
Fault status: Present in r0p1. Open.

Description

If the interrupt/event retention bits are non-zero in the IMP_CPUPWRCTLR_EL1 register to enable the FULL_RET power mode, then the power transition from ON to OFF/OFF_EMU might deadlock.

Configurations affected

All configurations are affected. Implementations are only affected if they clamp signals on the core power domain boundary when in the FULL_RET power mode.

Conditions

This erratum occurs under the following conditions:

1. The IMP_CPUPWRCTLR_EL1.WFI_RET_CTLR or IMP_CPUPWRCTLR_EL1.WFE_RET_CTLR fields are non-zero, enabling the FULL_RET power mode.
2. The core executes a WFI or WFE instruction, which causes the core to enter the FULL_RET power mode.
3. The implementation clamps the signals on the core power domain boundary as part of the FULL_RET entry.
4. There is traffic to the core that needs the core to transition back to the ON power mode. This traffic could be on the Utility Bus, the Debug APB interface, the GIC interface, or other external pins that are routed to the core.

Implications

If the erratum condition occurs, then the core will not leave the FULL_RET power mode, which will cause the system to deadlock.
Typically an implementation will only enable clamps on the core power domain boundary if it uses this mode to put the core logic into a retention state. If the FULL_RET mode is used for other low power techniques, for example only putting the RAMs into a retention state, then clamps may not be necessary on the core boundary.

Workaround

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

Category B (rare)

2986656
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. Fixed in r0p1.

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

2921482
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. Open

Description

A *Processing Element* (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 with LEGACY_TZ_EN set to 1.

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.

2928513
MPAM value associated with instruction fetch might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, 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.

2933585 L2D_CACHE_WB_CLEAN overcounts

Status

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

Description

Counting of the L2D_CACHE_WB_CLEAN event includes transfer of data directly to another 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 *Processing Element* (PE) processes a forwarding snoop from the DSU or 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.

2936120
Noncompliance with prioritization of Exception Catch debug events

Status

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

2940264
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. Fixed in r0p1.

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 under the following conditions:

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.

2940266
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. Fixed in r0p1.

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 under the following conditions:

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.

2963918
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. Fixed in r0p1.

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.

2982003
SPE latency counters are corrupted under certain conditions

Status

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

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.

3071658
TagMatch responses with error indication do not generate a SError abort

Status

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

Description

When tag checks are performed outside of the *Processing Element* (PE), the AMBA CHI protocol returns a TagMatch response that indicates whether or not the tag check succeeded or failed. If an error condition occurred while performing the tag check, the system might return the TagMatch response with an error indication. If this occurs, the PE should report a SError abort, but fails to do so.

Configurations affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

Conditions

This erratum occurs under the following conditions:

1. PE has *Memory Tagging Extension* (MTE) enabled in asynchronous checking of stores.
2. PE performs tag checked stores.
3. Write streaming causes the PE to send the stores to the interconnect as write transactions.
4. While performing the tag check operation for the write, the interconnect encounters an error condition while reading the tag value.

Implications

If the conditions are met, the interconnect might return a TagMatch response with an error indication, but the PE might not generate a SError abort. If the TagMatch response indicates a tag check failure (Resp=Fail), TFSR_ELx bits will still be updated.

Workaround

No workaround is required for this erratum.