Arm® Cortex-A520AE Core (MP187)

Software Developer Errata Notice

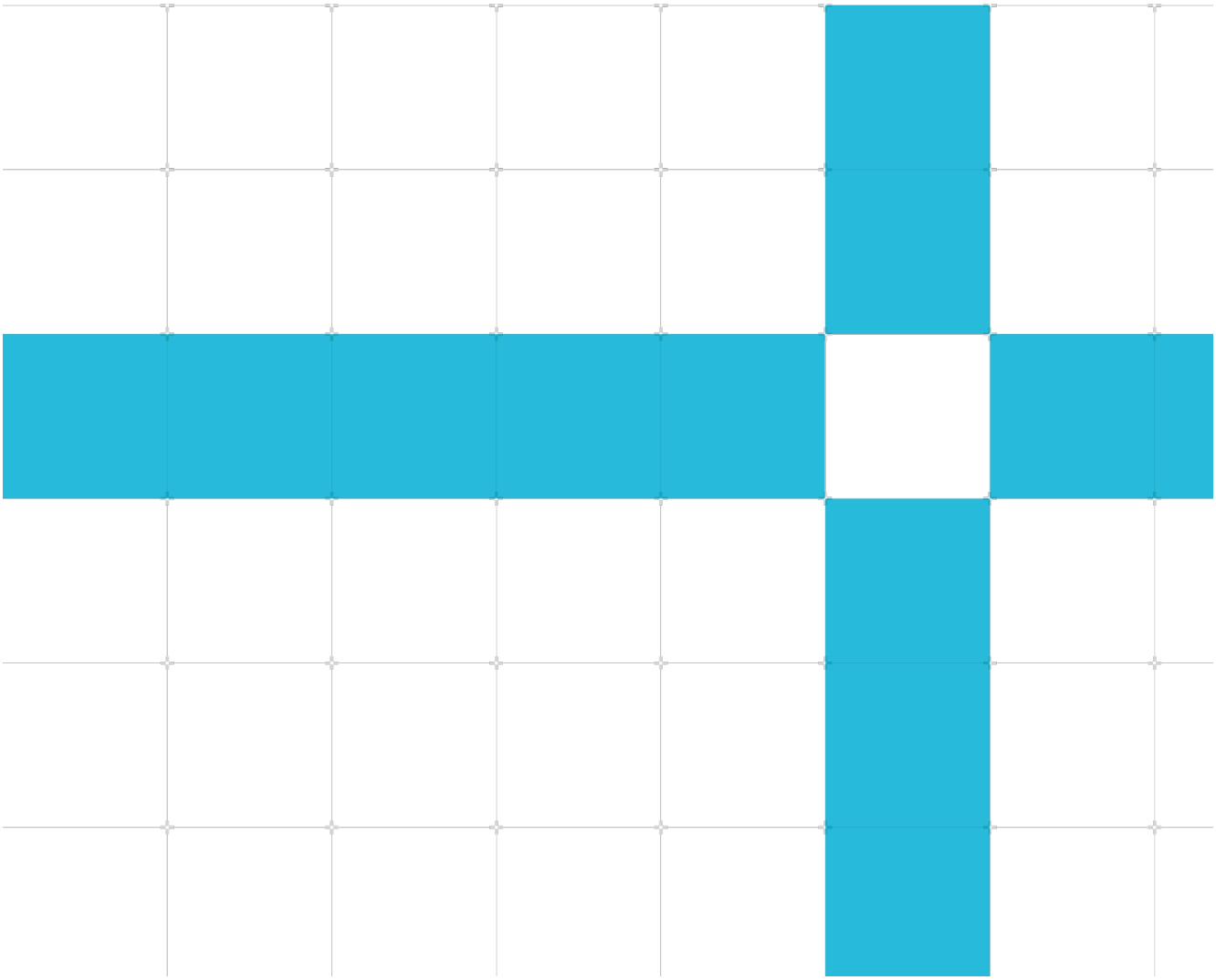
Date of issue: May 21, 2025

Non-Confidential

Document version: 5.0

Document ID: SDEN-2343003

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



This document (SDEN_2343003_5.0_en) was issued on May 21, 2025.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-2343003**

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**.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® Cortex-A520AE Core (MP187), 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.

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.

May 21, 2025: Changes in document version v5.0

ID	Status	Area	Category	Summary
3919721	New	Programmer	Category B	Power transition might deadlock on Utility Bus or APB access
3990231	New	Programmer	Category C	An ECC error detected during a core powerdown might result in deadlock

December 20, 2024: Changes in document version v4.0

ID	Status	Area	Category	Summary
3435249	Updated	Programmer	Category B	Unmodified MTE tags might be written back to memory
3802424	New	Programmer	Category B	Complex power transition might deadlock when a Utility bus or Debug APB access is sent to a core
3326983	Updated	Programmer	Category C	A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction
3414408	Updated	Programmer	Category C	The virtual address is not sign extended in EDWAR
3435248	Updated	Programmer	Category C	The core might lose register accesses or interrupts in low-power state
3470103	Updated	Programmer	Category C	Some PMU events do not count correctly
3627236	Updated	Programmer	Category C	External registers PMCEID2 and PMCEID3 are not implemented
3676732	Updated	Programmer	Category C	Interrupt signals generated by cores might be deasserted when in retention
3727550	New	Programmer	Category C	Performance might drop for core 1 in a 2-core complex configuration
3738915	New	Programmer	Category C	Consecutive CTI trigger events from the same ELA output might not be sent to the CTI
3786005	New	Programmer	Category C	CTI trigger events from a core in Standby state might not be sent correctly
3786500	New	Programmer	Category C	External debug request while transitioning to emulated off might cause core to deadlock
3791686	New	Programmer	Category C	Incorrect ESR_EL1.ISS.EX or ESR_EL2.ISS.EX for Software Step Exceptions
3817614	New	Programmer	Category C	Affinity ID info in Complex_RAS.ERRDEVAFF register is not correct in single-core complex configuration

September 10, 2024: Changes in document version v3.0

ID	Status	Area	Category	Summary
3435249	New	Programmer	Category B	Unmodified MTE tags might be written back to memory
3559262	New	Programmer	Category B	Power transition to FULL_RET is denied if FUNC_RET is also enabled
3672346	New	Programmer	Category B	CPU non-secure physical timer event interrupt might be triggered incorrectly
3674306	New	Programmer	Category B	Store operations might modify data twice
3711577	New	Programmer	Category B	CAS/CASP atomic instruction might get data corruption under certain condition
3326983	New	Programmer	Category C	A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction
3414408	New	Programmer	Category C	The virtual address is not sign extended in EDWAR
3435248	New	Programmer	Category C	The core might lose register accesses or interrupts in low-power state
3470103	New	Programmer	Category C	Some PMU events do not count correctly
3545700	New	Programmer	Category C	Power transition from EMU_OFF to OFF might not complete
3562078	New	Programmer	Category C	Load/Store instruction might get unexpected translation fault
3563909	New	Programmer	Category C	TRBE might record the wrong status/syndrome information in the TRBSR_EL1 register
3608624	New	Programmer	Category C	A Non-cacheable store exclusive instruction receiving an NDErr or DErr response might update memory and raise synchronous abort
3627235	New	Programmer	Category C	Some PMU events for Operation speculatively executed do not count correctly
3627236	New	Programmer	Category C	External registers PMCEID2 and PMCEID3 are not implemented
3650478	New	Programmer	Category C	Core in a dual core complex might complete power transition to OFF while a CTI trigger is in progress
3658235	New	Programmer	Category C	Some PMU events do not count correctly
3660306	New	Programmer	Category C	CPU might incorrectly wake up from WFE state due to a virtual event stream when in FULL_RET power mode
3675352	New	Programmer	Category C	Core might not execute some instructions in debug state after a reset catch debug event is generated
3676732	New	Programmer	Category C	Interrupt signals generated by cores might be deasserted when in retention
3685406	New	Programmer	Category C	The wrong shareability might be selected for instruction fetches when HCR_EL2.FWB == 1 while accessing a Non-shareable memory region
3715577	New	Programmer	Category C	IRG produces biased tag generation when GCR_EL1.RRND == 1

March 13, 2024: Changes in document version v2.0

No new or updated errata in this document version.

November 16, 2023: Changes in document version v1.0

No errata in this document version.

Errata summary table

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

ID	Area	Category	Summary	Found in versions	Fixed in version
3435249	Programmer	Category B	Unmodified MTE tags might be written back to memory	r0p0	r0p1
3559262	Programmer	Category B	Power transition to FULL_RET is denied if FUNC_RET is also enabled	r0p0, r0p1	Open
3672346	Programmer	Category B	CPU non-secure physical timer event interrupt might be triggered incorrectly	r0p0, r0p1	Open
3674306	Programmer	Category B	Store operations might modify data twice	r0p0, r0p1	Open
3711577	Programmer	Category B	CAS/CASP atomic instruction might get data corruption under certain condition	r0p0, r0p1	Open
3802424	Programmer	Category B	Complex power transition might deadlock when a Utility bus or Debug APB access is sent to a core	r0p0	r0p1
3919721	Programmer	Category B	Power transition might deadlock on Utility Bus or APB access	r0p0, r0p1	Open
3326983	Programmer	Category C	A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction	r0p0	r0p1
3414408	Programmer	Category C	The virtual address is not sign extended in EDWAR	r0p0	r0p1
3435248	Programmer	Category C	The core might lose register accesses or interrupts in low-power state	r0p0	r0p1
3470103	Programmer	Category C	Some PMU events do not count correctly	r0p0	r0p1
3545700	Programmer	Category C	Power transition from EMU_OFF to OFF might not complete	r0p0, r0p1	Open
3562078	Programmer	Category C	Load/Store instruction might get unexpected translation fault	r0p0, r0p1	Open
3563909	Programmer	Category C	TRBE might record the wrong status/syndrome information in the TRBSR_EL1 register	r0p0, r0p1	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3608624	Programmer	Category C	A Non-cacheable store exclusive instruction receiving an NDErr or DErr response might update memory and raise synchronous abort	r0p0, r0p1	Open
3627235	Programmer	Category C	Some PMU events for Operation speculatively executed do not count correctly	r0p0, r0p1	Open
3627236	Programmer	Category C	External registers PMCEID2 and PMCEID3 are not implemented	r0p0	r0p1
3650478	Programmer	Category C	Core in a dual core complex might complete power transition to OFF while a CTI trigger is in progress	r0p0, r0p1	Open
3658235	Programmer	Category C	Some PMU events do not count correctly	r0p0, r0p1	Open
3660306	Programmer	Category C	CPU might incorrectly wake up from WFE state due to a virtual event stream when in FULL_RET power mode	r0p0, r0p1	Open
3675352	Programmer	Category C	Core might not execute some instructions in debug state after a reset catch debug event is generated	r0p0, r0p1	Open
3676732	Programmer	Category C	Interrupt signals generated by cores might be deasserted when in retention	r0p0	r0p1
3685406	Programmer	Category C	The wrong shareability might be selected for instruction fetches when HCR_EL2.FWB == 1 while accessing a Non-shareable memory region	r0p0, r0p1	Open
3715577	Programmer	Category C	IRG produces biased tag generation when GCR_EL1.RRND == 1	r0p0, r0p1	Open
3727550	Programmer	Category C	Performance might drop for core 1 in a 2-core complex configuration	r0p0	r0p1
3738915	Programmer	Category C	Consecutive CTI trigger events from the same ELA output might not be sent to the CTI	r0p0, r0p1	Open
3786005	Programmer	Category C	CTI trigger events from a core in Standby state might not be sent correctly	r0p0, r0p1	Open
3786500	Programmer	Category C	External debug request while transitioning to emulated off might cause core to deadlock	r0p0, r0p1	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3791686	Programmer	Category C	Incorrect ESR_EL1.ISS.EX or ESR_EL2.ISS.EX for Software Step Exceptions	r0p0, r0p1	Open
3817614	Programmer	Category C	Affinity ID info in Complex_RAS.ERRDEVAFF register is not correct in single-core complex configuration	r0p0, r0p1	Open
3990231	Programmer	Category C	An ECC error detected during a core powerdown might result in deadlock	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

3435249
Unmodified MTE tags might be written back to memory

Status

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

Description

MTE tags for a location might be marked as dirty even without a STG* instruction modifying them, which might result in those MTE tags being written back to memory.

Configurations affected

This erratum affects platforms supporting MTE (parameter BROADCASTMTE = true) that try to re-use the 3% of MTE RAM as both tags and regular memory during the runtime of the system.

Conditions

This erratum occurs under the following conditions:

1. Memory location A is marked as Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. The core allocates location A into the L1 data cache or the L2 cache in Unique state with MTE tags. This allocation might be due to committed instructions, speculative execution, or data prefetching.
3. The core executes a store operation that does not modify MTE tags.
4. Another PE is requesting line A, or line A is naturally evicted. The core will provide the MTE tags for line A marked as dirty, but their value remains unchanged.

Implications

If the previous conditions are met, the MTE tags for memory location A might be marked as modified, and subsequently be written back, replacing the original MTE tag value.
When MTE tag memory is only used for tags and no external modification occurs, this is harmless.
If the Operating System re-uses the MTE tag memory as regular memory the tag write back might occur once the memory is in use as regular memory, causing data corruption.

Workaround

If the interconnect always fetches tags, regardless of the CPU memory attributes:

- No re-use of the MTE tag memory is possible. This memory should not be described in firmware tables.

If the interconnect only fetches tags for memory that is marked as tagged, when changing the use of a page of memory from MTE tags to regular memory, the OS should:

- Remove all tagged mappings for the target page (data page corresponding to the tag page being re-used) to prevent fetching of data as tags.
- Clean and Invalidate the target page, via DC IGVAC, to remove any tags that have been cached.

3559262
Power transition to FULL_RET is denied if FUNC_RET is also enabled

Status

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

Description

A power transition to FULL_RET will be incorrectly denied if FUNC_RET is also enabled.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core has non-zero bits in the IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL or WFE_RET_CTRL fields, enabling the FULL_RET power mode.
2. The core has non-zero bits in the IMP_CPUPWRCTLR_EL1.VPU_PWR_CTRL field, enabling the FUNC_RET power mode.
3. The core PPU requests a transition to FULL_RET.

Implications

If the previous conditions occur, the power transition to FULL_RET will be denied, and the core will not enter into FULL_RET power mode. This means that only one of the two power modes can be in use at any time, limiting the amount of power savings that are achievable.

Workaround

These modes are disabled by default. Firmware should enable only FUNC_RET or FULL_RET, but not both. The choice of mode will depend on whether FUNC_RET or FULL_RET gives greater power savings, which will depend on the implementation and expected workloads.

3672346
CPU non-secure physical timer event interrupt might be triggered incorrectly

Status

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

Description

The **nCNTPNSIRQ** interrupt might be triggered at an incorrect time when the core is in FULL_RET mode.

Configuration affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. EL2Enabled() i.e. SCR_EL3.{NS, EEL2} is not {0, 0}.
2. SCR_EL3.ECVEn is 1,
3. CNTHCTL_EL2.ECV is 1.
4. HCR_EL2.{E2H, TGE} is not {1, 1}
5. CNTP_CTL_EL0.{IMASK, ENABLE} is set {0,1}
6. The core executes a WFI, WFIT, WFE, or WFET instruction and enters the FULL_RET power mode.

Implications

If the conditions are met, then

- CNTPOFF_EL2 is ignored and is not applied to the PhysicalCountInt() to derive virtual counter value that is used to trigger nCNTPNSIRQ interrupt.
- **nCNTPNSIRQ** interrupt might be triggered at an incorrect time.

Workaround

Disable entering full retention mode by setting both IMP_CPUPWRCTLR_EL1.{WFE_RET_CTRL} and IMP_CPUPWRCTLR_EL1.{WFI_RET_CTRL} to 3'b000.

3674306
Store operations might modify data twice

Status

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

Description

Non-L1 allocating store instructions of less than four Memory Tagging Extension (MTE) tags for the same cacheline, with full cacheline data update to the same cacheline, might modify data twice.

Configurations affected

This erratum affects configurations with two cores in a complex.

Conditions

The erratum occurs under the following conditions:

1. Cacheline X is cached in the complex in Unique state, without MTE tags.
2. Processing Element (PE) A in the same complex executes a number of store instructions that collectively modify all the data of cacheline X but not all MTE tags for X.
3. The stores are gathered and do not allocate in the L1 cache.
4. PE B outside the complex does a store to cacheline X.
5. Unlikely timing and micro-architecture conditions occur.

Implications

If the conditions are met, a PE observing X might see the cacheline

- first with the value written by PE A
- then with the value written by PE B
- again with the value written by PE A

Workaround

Write streaming can be disabled for MTE stores by setting IMP_CPUACTLR_EL1.MTEALLCWSDIS to 0b1, for example using the following code sequence

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #27 #1
MSR S3_0_C15_C1_0, x0
```

3711577
CAS/CASP atomic instruction might get data corruption under certain condition

Status

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

Description

A CAS/CASP atomic instruction executed as near atomic instruction might get silent data corruption under certain condition.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core executes an atomic CAS/CASP instruction.
2. The previously mentioned atomic instruction hits against an ongoing linefill to the same cache line as the CAS/CASP instruction.
3. Unlikely timing and micro-architectural conditions occur.

Implications

If the previous conditions are met, the atomic CAS/CASP might never perform the write even in case of successful comparison, causing the CAS/CASP data lost.

Workaround

To prevent this erratum from occurring, software can set IMP_CPUACTLR_EL1[9] = 1, for example using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #9, #1
MSR S3_0_C15_C1_0, x0
```

3802424
Complex power transition might deadlock when a Utility bus or Debug APB access is sent to a core

Status

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

Description

A power transition or warm reset on both cores in a complex might deadlock when there is an access on the Utility bus or Debug Advanced Peripheral Bus (APB) interface during the transitions.

Configurations affected

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

Conditions

The erratum occurs under the following conditions:

- One core in the complex is transitioning from the OFF or OFF_EMU power mode to ON.
- The other core in the complex is either performing a warm reset requested by the Reset Management Register (RMR_ELx), or is also transitioning from the OFF or OFF_EMU power mode to ON.
- An access is made on the Utility bus or Debug APB interface to a memory mapped register in one of the cores, while the power transitions are in progress.

Implications

When the previous conditions occur, the complex might deadlock. The power transition will not complete, and the Utility bus or Debug APB transaction will not complete. This erratum will have a negligible effect on the metrics within the FMEDA report.

Workaround

When the system cannot prevent Utility bus and Debug APB accesses to cores that have not yet reached the ON power mode, then it must restrict the power transitions that can occur on the two cores in parallel.

- In case of multiple cores power transitioning to ON at the same time, firmware can control the power transitions using the LOCK feature (PPU_PWPR.LOCK_EN) in the core's Power Policy Unit

(PPU) such that only one core of the complex can transition to ON at a time.
- When EL3 firmware must make use of RMR_EL3, it must coordinate with an external agent to ensure a core reset due to RMR_EL3.RR does not occur at the same time as a power transition for another core in the complex.

3919721
Power transition might deadlock on Utility Bus or APB access

Status

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

Description

A power transition might deadlock if there is an access to a core register on the Utility Bus or debug APB interface during the transition.

Configurations affected

This erratum affects all configurations of the core. However it is only realistic to hit all the conditions on configurations with no L2 cache.

Conditions

The erratum occurs under the following conditions:

1. The core power transitions from ON to OFF, or OFF_EMU to OFF.
2. An access is made on the Utility Bus or debug APB interface to access a memory mapped register in the core during the ON to OFF power transition, or an access is made on the debug APB interface to access a memory mapped register in the core during the OFF_EMU to OFF power transition. For Utility bus accesses, the address must be in the range 0x<n>9_0000 to 0x<n>F_FFFF where <n> is the core number.
3. The Utility Bus or debug APB access must start before the complex power transition starts, but because of delays on the internal buses of the cluster, the transaction does not complete until after the end of the complex power transition. The complex power transition includes flushing of the L2 cache which will take many hundreds of cycles. It is implausible for the transaction to be delayed this long, therefore this condition is only realistic on a configuration that has no L2 cache.

Implications

If the previous conditions occur, the Utility Bus or APB access might not complete, leading to a system deadlock.

Workaround

There is a workaround that avoids the problem but might not be possible to apply in many systems. There is a second partial workaround that does not cover all cases, but can be used if the first workaround is not suitable. Only one of these two options should be applied:

1. If the system and software can ensure that no Utility Bus or APB accesses can be made during a powerdown sequence, then this will avoid the erratum.
2. The system component that is programming the Power Policy Units (PPUs), typically a System Control Processor (SCP), should ensure that any core power transition from ON to OFF is replaced by the following sequence: ON to OFF_EMU to OFF. This will prevent Utility Bus accesses from causing the issue, but will not prevent APB accesses from causing a problem, therefore the problem can still occur during debug.
   If the PPUs are being used in static mode, then the SCP that is requesting the transition can request the additional transition in the sequence.
   If the PPUs are being used in dynamic mode, then the following sequence will ensure that all transitions to OFF power mode are made using the OFF_EMU mode:
   - Set the PPU_PWPR.LOCK_EN bit if it is not already set.
   - Set PPU_PWPR.PWR_POLICY to OFF_EMU instead of OFF.
   - When the PPU reaches the OFF_EMU mode, the PPU_PWPR.PWR_POLICY field should be reprogrammed to OFF. Either of the LOCKED_IRQ_MASK or EMU_ACCEPT_IRQ_MASK bits in the PPU_IMR register can be cleared to request an IRQ to be generated so that the SCP knows when this mode is reached.
   - The PPU_PWCR.PWR_DEVACTIVEEN field should be cleared to ensure that a new wakeup event arriving does not prevent the transition to OFF.
   - The SCP should wait for the PPU to reach OFF.
   - Once the PPU has reached OFF, the PPU_PWCR.PWR_DEVACTIVEEN field can be restored to its previous value.
   - The PPU_PWPR.PWR_POLICY should be set back to OFF_EMU. This can be done either as soon as the PPU has reached OFF, or it can wait until immediately before the PPU_UNLK register is written when the core is requested to power on again.

Category B (rare)

There are no errata in this category.

Category C

3326983
A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction

Status

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

Description

A watchpoint event might not be generated correctly for a tag-checked SVE non-fault or first-fault load instruction.

Configurations affected

This erratum affects configurations with BROADCASTMTE = true.

Conditions

This erratum occurs under the following conditions:

1. MTE checking is enabled (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn != 0b00).
2. The core executes a checked SVE non-fault or first-fault load instruction.
3. The load instruction crosses a 16-byte boundary, but does not cross a page boundary.
4. The load above fails the MTE tag check on the 16-byte-aligned quantity that crossed over the 16-byte boundary.
5. Watchpoints are enabled, and set to a memory location covered by the load above. For a first-fault load, the watchpoint is not covering the first accessed element.
6. Uncommon micro-architectural conditions occur.

Implications

If the conditions are met, the watchpoint event might not be generated, and the FFR register might not reflect that a watchpoint event occurred.

Workaround

No workaround is required for this erratum.

3414408
The virtual address is not sign extended in EDWAR

Status

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

Description

EDWAR returns the virtual data address being accessed when a Watchpoint Debug Event was triggered. Virtual addresses are 64-bit in AArch64.

Virtual addresses reported in EDWAR[63:49] get truncated and are padded with zeros instead of sign extended.

The valid VA range for each TTBR is maximum of 48-bits (this core does not support FEAT_LVA), which is why VA was getting microarchitecturally truncated to save some flops.

In EL1+EL0 and EL2+EL0 translation regimes, there are two TTBRs. Addresses in the TTBR1 range are 0xFFFFFFFFFFFFFFFF down to the size of the TTBR range (so down to 0xFFFF000000000000 if using the full 48-bit range).

With the truncation+ZeroPadding, if there is a watchpoint on the address 0xFFFF000000000000, then EDWAR will report 0x000100000000000, which violates the architecture.

Configurations affected

This erratum affects all configurations.

Conditions

1. Watchpoint is enabled by DBGWCR<n>_EL1 register.
2. Virtual Address in DBGWVR<n>_EL1 belongs to TTBR1 range.

Implications

Virtual address reported in EDWAR[63:49] will be incorrect.

Workaround

After reading EDWAR, Debugger can perform something as below:

```
if(EDWAR[48]==1) {
```

```
    EDWAR = EDWAR | 0xFFFE000000000000;
}
```

3435248
The core might lose register accesses or interrupts in low-power state

Status

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

Description

In rare circumstances, a core in low-power state might lose debug accesses, system register accesses or interrupts. In addition, microarchitectural accesses related to reset might be lost. This might lead to a system deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. At least one core in a complex is in the low-power state or Off power mode.
2. At least one other component in the system is active and driving system register or interrupt traffic through the complex.
3. One of the following events occurs, targeting the core in low-power state:
    - Warm or power-on reset
    - External Debug register access
    - System register access through the Utility bus
    - An interrupt
4. Uncommon, timing-sensitive microarchitectural conditions occur.

Implications

If the conditions are met, then the core in low-power state might have its clock automatically gated during any of the mentioned events, which might lead to a system deadlock.
The timing-sensitive conditions are expected to be rare. The system will recover in most cases when a subsequent event causes the clock to be restored, allowing the first event to complete, resolving the deadlock.

Workaround

Due to the rarity of the conditions, and the potential for a subsequent event to break the deadlock, Arm does not believe that a workaround is required.

3470103
Some PMU events do not count correctly

Status

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

Description

Some PMU events have notable inaccuracies. Where applicable, any consistent architectural inaccuracies have been described. Otherwise, the inaccuracies are inconsistent or occur due to microarchitectural conditions.

- 0x0006 LD_RETIRED - counts atomic instructions which do not return data to the PE, such as STADD. Missing atomic instructions which return data, such as LDADD and CAS.
- 0x0007 ST_RETIRED - missing atomic instructions which do not return data and DC ZVA instructions
- 0x000B CID_WRITE_RETIRED - counts reads of CONTEXTIDR_EL1
- 0x006F STREX_SPEC
- 0x0070 LD_SPEC - counts some system register reads
- 0x0071 ST_SPEC - counts some system register writes, barriers, and cache maintenance operations

Configurations affected

This erratum affects all configurations.

Conditions

The conditions will depend on the event being counted.

Implications

Some PMU events will give inaccurate results.

Workaround

There are no workarounds for these events. ARM believes it is unlikely that a workaround would be necessary. If the precise accuracy of the mentioned events is a concern, please contact ARM.

3545700
Power transition from EMU_OFF to OFF might not complete

Status

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

Description

If debug operations are made to a core that is transitioning from EMU_OFF to OFF power mode, then the transition might not complete.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core is in the EMU_OFF power mode
2. PPU_PWPR.PWR_POLICY is re-programmed to bring this core to OFF power mode (from 0b0001 to 0b0000)
3. The external debugger makes an access to the core
4. Additional microarchitectural timing conditions occur

Implications

If the previous conditions are met, the power transition to OFF will not complete.

Workaround

This erratum is not expected to be seen in production devices as the expected use case is during development stages. If required, debugger can re-program PPU_PWPR.PWR_POLICY to OFF state after debug operations are completed.

3562078
Load/Store instruction might get unexpected translation fault

Status

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

Description

A load/store instruction might get an unexpected translation abort when executed together with a PRFM instruction or a mis-predicted non-fault load instruction, when the TCR_ELx.NFDy bit is set.

Configurations affected

This erratum affects all configuration.

Conditions

This erratum occurs if one of the 2 following sequences happen.
First possible sequence of conditions:

1. Core running in EL0
2. The TCR_ELx.NFDy bit is set
3. PRFM instruction followed by Load/Store/CMO instruction which shares the same translation as the PRFM instruction
4. Uncommon micro-architecture condition happens

Second possible sequence of conditions:

1. Core running in EL0
2. The TCR_ELx.NFDy bit is set
3. Unaligned Load/Store instruction cross page boundary, followed by conditional or unconditional branch instruction
4. followed by a mis-predicted SVE non-fault contiguous load instruction or PRFM instruction which share the same translation as the above mentioned Load/Store
5. Uncommon micro-architecture condition happens

Implications

If the previous conditions are met, the Load/Store/CMO instruction might generate unexpected L0 level translation fault.

Workaround

There is no workaround required for this erratum.

3563909
TRBE might record the wrong status/syndrome information in the TRBSR_EL1 register

Status

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

Description

The Trace Buffer Unit might record the wrong status/syndrome information in TRBSR_EL1 register when a translation abort happens.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The Trace Buffer is enabled
2. TRBLIMITR_EL1.FM !=2'b11
3. The trace buffer writes to a VA that encounters a translation fault

Implications

If the previous conditions are met two situations can occur, with TRBSR_EL1.wrap bit set even if no write data wrap event happens in both cases:

- The status/syndrome information recorded in the TRBSR_EL1 register might not correspond to the translation fault.
- The TRBSR_EL1 register record as translation fault.

The TRBPTR_EL1 Trace Buffer Write Pointer Register still records the correct page where the translation fault happens.

Workaround

ARM does not expect the condition 3 to happen, no workaround is required for this erratum.

3608624
A Non-cacheable store exclusive instruction receiving an NDErr or DErr response might update memory and raise synchronous abort

Status

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

Description

A Non-cacheable store exclusive instruction receiving an NDErr or DErr response will raise a synchronous abort, but it might also update the memory.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. Store exclusive instruction to shareable memory other than Inner-Writeback and Outer-Writeback.
2. The write comp response for this store exclusive instruction returned with an NDerr or Derr error response.

Implications

If the previous conditions are met, the store exclusive instruction will raise synchronous abort, but the memory might also be updated by the store exclusive instruction.

Workaround

There is no workaround required.

3627235
Some PMU events for Operation speculatively executed do not count correctly

Status

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

Description

PMU event 0x0073 DP_SPEC incorrectly does not include the following instruction types:

- System register instructions
- System instructions
- Hint instructions
- PAC instructions

PMU event 0x0071 ST_SPEC incorrectly includes some memory related barriers, system registers, and system instructions.

PMU event 0x0076 PC_WRITE_SPEC has the following inaccuracies:

- Includes ISB instructions while 0x000C PC_WRITE_RETIRED does not. The inclusion of ISB in both events is implementation defined. These two events should be consistent, and the intention was that ISB would not be included.
- System register writes which include a Context Synchronization Event (such as direct writes to PSTATE) incorrectly count this event.
- ERET instructions incorrectly do not count this event.

Configurations affected

This erratum affects all configurations.

Conditions

No specific conditions are needed for this erratum.

Implications

Extra consideration should be taken when comparing PMU results for these events with other CPUs. The overall impact should be low because barriers, system register instructions, system instructions, and hints are expected to be a low percentage of code. The effect of PAC instructions being missing from DP_SPEC could have slightly higher impact, depending on the routine being measured.

Workaround

There is no workaround for this erratum. A workaround should not be necessary.

3627236
External registers PMCEID2 and PMCEID3 are not implemented

Status

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

Description

External registers PMCEID2 and PMCEID3 should be accessible using the external interface to the Performance Monitors registers. These registers should be mapped as follows to System registers:

- PMCEID2 to PMCEID0_EL0[63:32]
- PMCEID3 to PMCEID1_EL0[63:32]

Due to this erratum, reads of these external registers always return 0.

Configurations affected

This erratum affects all configurations.

Conditions

An external read of PMCEID2 or PMCEID3 occurs.

Implications

These registers are read-only and indicate whether certain *Performance Monitoring Unit* (PMU) events are implemented, using one bit per event. Reading a 1 indicates the event is implemented. The following events are implemented, but their corresponding bit will read as 0:

- 0x4005 STALL_BACKEND_MEM
- 0x4006 L1I_CACHE_LMISS
- 0x400C TRB_WRAP
- 0x400E TRB_TRIG
- 0x4010 TRCEXTOUT0
- 0x4011 TRCEXTOUT1
- 0x4012 TRCEXTOUT2
- 0x4013 TRCEXTOUT3
- 0x4018 CTI_TRIGOUT4
- 0x4019 CTI_TRIGOUT5
- 0x401A CTI_TRIGOUT6
- 0x401B CTI_TRIGOUT7

- 0x4020 LDST_ALIGN_LAT
- 0x4021 LD_ALIGN_LAT
- 0x4022 ST_ALIGN_LAT
- 0x4024 MEM_ACCESS_CHECKED
- 0x4025 MEM_ACCESS_RD_CHECKED
- 0x4026 MEM_ACCESS_WR_CHECKED

The following events are only implemented in certain configurations. In configurations where they are present, they are affected by this erratum:

- 0x4009 L2D_CACHE_LMISS_RD
- 0x400B L3D_CACHE_LMISS_RD

Workaround

The correct information can be accessed through the System registers PMCEID0_EL0 and PMCEID1_EL0, and is documented in the *Technical Reference Manual* (TRM).

3650478
Core in a dual core complex might complete power transition to OFF while a CTI trigger is in progress

Status

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

Description

A core in dual core complex might complete a power transition to OFF while a CTI trigger is in progress, which might cause a deadlock.

Configurations affected

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

Conditions

The erratum occurs under the following conditions:

1. The CTIs are enabled to generate triggers
2. A core makes one of the following power mode transitions:
    - ON to OFF
    - OFF_EMU to OFF
3. The CTI generates a trigger that targets at least both cores in the complex

Implications

If the previous conditions occur, the core might power transition to OFF while a CTI trigger is in progress. This will prevent transactions from completing on the debug APB interface, which might limit the debug capability on the system or cause a system deadlock.

Workaround

When cross triggers are in use for debugging, then the core can program DBGPRCR_EL1.CORENPDRQ to 1'b1 so that the core enters into the OFF_EMU power mode rather than the OFF mode.

3658235
Some PMU events do not count correctly

Status

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

Description

Some PMU events have notable inaccuracies. Where applicable, any consistent architectural inaccuracies have been described. Otherwise, the inaccuracies are inconsistent or occur due to microarchitectural conditions.

- 0x0001 L1I_CACHE_REFILL - counts for cache misses while instruction cache is disabled
- 0x0070 LD_RETIRED - counts atomic instructions which do not return data to the PE, such as STADD. Missing atomic instructions which return data, such as LDADD and CAS
- 0x0071 ST_RETIRED - missing atomic instructions which do not return data and DC ZVA instructions
- 0x00EE IMP_STALL_SLOT_BACKEND_ILOCK - counts stalls caused by interlock after dispatch instead of interlock cycles per slot at dispatch
- 0x8164 STALL_BACKEND_MEMBOUND - counts when there are no micro-operations available to dispatch

Configurations affected

This erratum affects all configurations.

Conditions

The conditions will depend on the event being counted.

Implications

Some PMU events will give inaccurate results.

Workaround

There are no workarounds for these events. ARM believes it is unlikely that a workaround would be necessary. If the precise accuracy of the mentioned events is a concern, please contact ARM.

3660306
CPU might incorrectly wake up from WFE state due to a virtual event stream when in FULL_RET power mode

Status

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

Description

Core incorrectly wakes up from WFE state due to a virtual event stream.

Configuration affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. The pseudocode function EL2Enabled() is TRUE, i.e. SCR_EL3.NS == '1' or SCR.EL3.EEL2 == '1'
2. HCR_EL2.<E2H,TGE> == '11'
3. CNTKCTL_EL1.EVNTEN == '1'
4. The core executes a WFE or WFET instruction and enters the FULL_RET power mode.

Implications

If the conditions are met, then the virtual event stream using the values of the CNTKCTL_EL1.{EVNTEN, EVNTDIR, EVNTI, EVNTIS} fields is enabled to wakeup the core from WFE or WFET state.

Workaround

No workaround is expected to be required for this erratum. If one is needed, disable entering full retention mode by setting both IMP_CPUPWRCTLR_EL1.WFE_RET_CTRL and IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL to '000'.

3675352
Core might not execute some instructions in debug state after a reset catch debug event is generated

Status

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

Description

Under certain microarchitectural conditions, a *Processing Element* (PE) might not execute some instructions pushed via an external debugger to the EDITR. An external debugger might also observe a corrupted EDSCR.STATUS value.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs after the following sequence:

1. Halting debug is not allowed.
2. The core exits reset state and a reset catch debug event is generated.
3. Halting debug is allowed.
4. The core enters debug state for another reason.
5. Certain microarchitectural conditions apply.

Implications

Under the above conditions, instructions pushed by an external debugger to the core via the EDITR might not be executed. The EDSCR.STATUS value might also appear corrupted.

Workaround

No workaround is expected to be required for this erratum.

3676732
Interrupt signals generated by cores might be deasserted when in retention

Status

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

Description

If a core enters the FULL_RET power mode while one of the interrupt outputs is asserted, the interrupt output might be deasserted without the interrupt having been acknowledged.

Configuration affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. One of the following interrupt outputs generated by the core is asserted:
    - **nTBEIRQ**
    - **nPMUIRQ**
    - **nCOMMIRQ**
    - **nVCPUMNTIRQ**
    - **nCOREERRIRQ**
    - **nCOREFAULTIRQ**
2. The core executes a **`WFI`**, **`WFIT`**, **`WFE`**, or **`WFET`** instruction and enters the FULL_RET power mode.

Implications

If the conditions are met, then the asserted interrupt outputs might become deasserted while in the FULL_RET power mode. Once the core exits FULL_RET, the interrupt output will be asserted again. If the interrupt is relied upon to wake the core from the **`WFI`**, **`WFIT`**, **`WFE`**, or **`WFET`** instruction, and no other wake-up events occur, the core might never wake up.

Workaround

There is no workaround.

3685406 The wrong shareability might be selected for instruction fetches when HCR_EL2.FWB == 1 while accessing a Non-shareable memory region

Status

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

Description

In certain architectural states that include HCR_EL2.FWB == 1, and accessing Non-shareable memory region, the core might use the wrong shareability for instruction fetch accesses.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following conditions are met:

- Core is running in EL1 or EL0 and using EL1&0 translation regime
- Stage 1 and Stage 2 MMU are on
- EL2 is enabled
- HCR_EL2.{FWB,ID,CD} = {1,0,0}
- SCTLR_EL1.{I,C} = {0,0}
- The combined S1 and S2 shareability is Non-shareable

Implications

If the conditions of the erratum are met, instruction accesses will use Non-shareable attributes while the data accesses will use Outer Shareable attributes. However, Arm does not expect hypervisors to configure memory as Non-shareable at Stage 2. Hence the implications of the erratum are not expected to be problematic.

Workaround

No workaround is expected to be required for this erratum.

3715577
IRG produces biased tag generation when GCR_EL1.RRND == 1

Status

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

Description

The algorithm used by the IRG instruction when GCR_EL1.RRND == 1 is biased and might generate tags from a not uniformly random distribution when certain exclusion masks are being used.
When GCR_EL1.RRND == 0, the IRG instruction generates tags following the ChooseNonExcludedTag() algorithm which is also biased as it was intended for debugging purposes.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following conditions are met:

1. GCR_EL1.RRND == '1'
2. The core executes an IRG instruction.

Implications

If the conditions mentioned above are met, random tag allocation might be biased. Arm does not consider the risk of a bias in the generation of random tags to change the security guarantees and value of MTE, or have any practical security impact.

Workaround

No workaround is expected to be required for this erratum.

3727550
Performance might drop for core 1 in a 2-core complex configuration

Status

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

Description

Performance might drop for core 1 in a 2-core complex configuration.

Configurations affected

This erratum affects all configurations with a 2-core complex, and the L2 cache is present (parameter L2_CACHE set to True).

Conditions

The erratum occurs under the following condition:

- Software runs on core 1 in a 2-core complex configuration

Implications

The performance of software running on core 1 might have a noticeable performance drop compared with software running on core 0. The performance drop depends on the software workload. Arm observed a 2.5% performance drop on core 1 for a Specint2k17 run.

Workaround

No workaround is available.

3738915
Consecutive CTI trigger events from the same ELA output might not be sent to the CTI

Status

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

Description

When consecutive *Cross Trigger Interface* (CTI) trigger events from the same *Embedded Logic Analyzer* (ELA) output occur, the core will only send the first one to the CTI. New trigger events from the same ELA output won't be sent until a different CTI input trigger event or a debug event occurs, or the core is reset.

The CTI input trigger events are as follows:

- Cross-halt
- *Performance Monitoring Unit* (PMU) counter overflow
- 4 *Embedded Trace Extension* (ETE) trace unit external outputs
- 2 ELA outputs

The type of debug events are as follows:

- The processor leaves cold reset.
- The processor indicates to the External Debugger that the core has entered or left debug state.

Configurations affected

This erratum only affects configurations that include ELA.

Conditions

The erratum occurs under the following conditions:

- ELA is programmed to drive one of its trigger outputs
- The trigger event occurs twice, without any intervening CTI input trigger events or debug events or reset.

Implications

Triggers are used to perform debug actions such as entering halting debug mode or starting trace. Missing triggers will prevent these actions from happening, without any further implications for future triggers or debug actions.

Workaround

No workaround is expected to be required for this erratum as the impact is limited to debug only, and is specifically limited to ELA, which should not be present in production devices.

3786005
CTI trigger events from a core in Standby state might not be sent correctly

Status

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

Description

The debug and trace events which are connected to *Cross Trigger Interface* (CTI) input triggers might not be sent, be delayed, or be sent multiple times while a core is entering Standby state.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. CTI is enabled by setting CTICONTROL.GLBEN to 1.
2. The core is entering Standby state after executing one of the WFI, WFE, WFIT, or WFET instructions.
3. During the transition to Standby state, one of the following affected trigger events occurs with microarchitectural timing conditions:
    - *Performance Monitoring Unit* (PMU) counter overflow
    - *Embedded Trace Extension* (ETE) trace external output
    - *Embedded Logic Analyzer* (ELA) trigger event

Implications

Under different microarchitectural timing conditions one of the following can occur:

1. A trigger might get delayed until the core exits Standby state
2. In the case of ELA trigger, it might not be observed at all
3. The trigger event might be observed multiple times, which might cause deadlocks

Triggers can be used to request debug actions such as halting mode or starting instruction trace. Missing or delaying triggers mainly affects those debug actions. All of these implications are mitigated by typically low core activity after a Standby instruction, as all trigger events are in response to some core activity.

The microarchitectural conditions for the third implication are particularly rare. The impact of observing the same trigger multiple times in rapid succession on debug actions is low, as the typical delay in responding to a trigger will mean the action is performed only once, and the duplicate triggers will have no effect. However, it is possible that the multiple triggers in flight could cause a deadlock between other types of traffic or power transitions.

Workaround

No workaround is expected to be required for this erratum.

3786500
External debug request while transitioning to emulated off might cause core to deadlock

Status

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

Description

When an external debug request is sent to a core that is transitioning to emulated off power mode, the transition to emulated off might not complete and any following requests for power mode transitions will be blocked. In this state, the core will not wake from standby for any reason nor enter halting debug mode.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following sequence of conditions apply.

1. Either of the following is performed to prepare for entry to emulated off power mode:
   - Dynamic entry to emulated off is enabled using the PPU_PWPR register in the core *Power Policy Unit* (PPU)
   - The core requests entry to emulated off by writing to the DBGPRCR_EL1 and IMP_CPUPWRCTLR_EL1 registers
2. The core finishes preparing for powerdown and executes a Standby instruction: WFI, WFE, WFIT, WFET.
3. External debug request arrives at the core under microarchitectural timing conditions, before the transition to emulated off completes.

Implications

When the conditions are met, there is no way to recover from the hung state.

Workaround

No workaround is expected to be required for this erratum as emulated off is not expected to be used in production devices.

3791686
Incorrect ESR_EL1.ISS.EX or ESR_EL2.ISS.EX for Software Step Exceptions

Status

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

Description

After Software-Stepping over a Load exclusive instruction, and under certain micro-architectural conditions, the core might report ESR_EL1.ISS.EX or ESR_EL2.ISS.EX as 0x0 instead of 0x1.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum can occur if all of the following conditions apply:

- Conditions to enter Software-Step state:
  - MDSCR_EL1.SS = 1
  - SPSR_ELx.SS = 1
  - ELR_ELx points to an instruction being stepped
- ERET to enter Software-Step execution state.
- One of the following Load exclusive instructions is executed in Software-Step mode:
  - LDXP
  - LDXR
  - LDXRB
  - LDXRH

- Timing-sensitive micro-architectural conditions occur.

Implications

Under the conditions previously defined, the core might report ESR_EL1.ISS.EX or ESR_EL2.ISS.EX as 0x0 instead of 0x1.

Workaround

No workaround is expected to be required for this erratum.

3817614
Affinity ID info in Complex_RAS.ERRDEVAFF register is not correct in single-core complex configuration

Status

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

Description

The Complex_RAS.ERRDEVAFF register value is not correct in single-core complex configurations.

Configurations affected

This erratum affects all configurations with a single-core complex.

Conditions

The erratum occurs if the following condition applies:

- Read of Complex RAS node ERRDEVAFF register

Implications

Reading the Complex RAS node ERRDEVAFF register returns the ERRDEVAFF.F0V bit to be 0, indicating ERRDEVAFF.Aff0 is not valid. This is not correct for a single-core complex configuration.

Note, with the deprecation of this register in the Advanced Configuration and Power Interface (ACPI) for the Armv8-A specification, the topology is described in firmware tables instead.

Workaround

There is no workaround required for this erratum.

3990231
An ECC error detected during a core powerdown might result in deadlock

Status

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

Description

If an ECC error is detected in the L1 data cache during a core powerdown the ongoing core powerdown might fail to complete causing a deadlock.

Configurations affected

All configurations with CORE_CACHE_PROTECTION set to TRUE are affected.

Conditions

The erratum might occur if all the following conditions apply:

- Core A is transitioning to OFF or OFF_EMU state.
- Core B executes a memory operation for address X which is present in core A.
- Timing sensitive micro-architectural conditions occur including an ECC error for address X in core A.

Implications

If the conditions described are met the core powerdown might fail to complete leading to a deadlock.

Workaround

There is still substantial benefit being gained from the *Error Correcting Code* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum.

If a workaround is necessary, a software-based flush of the L1 data cache can be completed before executing the final WFI for powerdown to avoid this issue. An example routine for flushing the L1 data cache is provided below.

```
// Select L1 Data Cache
    MOV     X0, #0
    MSR     CSSELR_EL1, X0
    ISB

// Read the cache size ID register
    MRS     X0, CCSIDR_EL1
```

```
// Number of sets = CCSIDR[55:32]
    LSR     X2, X0, #32
    AND     X2, X2, #0xFFFFFF
    ADD     X2, X2, #1

// Number of ways = CCSIDR[23:3]
    AND     X3, X0, #0xFFFFFF
    LSR     X3, X3, #3
    ADD     X3, X3, #1

// Loop over all ways and sets
    MOV     X0, #0
WayLoopStart:
    MOV     X1, #0
    LSL     X4, X0, #30
SetLoopStart:
    DC      CISW, X4

    ADD     X1, X1, #1
    ADD     X4, X4, #0x40
    CMP     X1, X2
    B.NE    SetLoopStart

    ADD     X0, X0, #1
    CMP     X0, X3
    B.NE    WayLoopStart

Done:
    DSB     SY
    ISB
```

Proprietary notice

(PRE-1121-V1.0)

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Product revision status

The rxpy identifier indicates the revision status of the product described in this manual, where:

rx

Identifies the major revision of the product.

py

Identifies the minor revision or modification status of the product.