Arm® C1-Ultra (MP201)

Software Developer Errata Notice

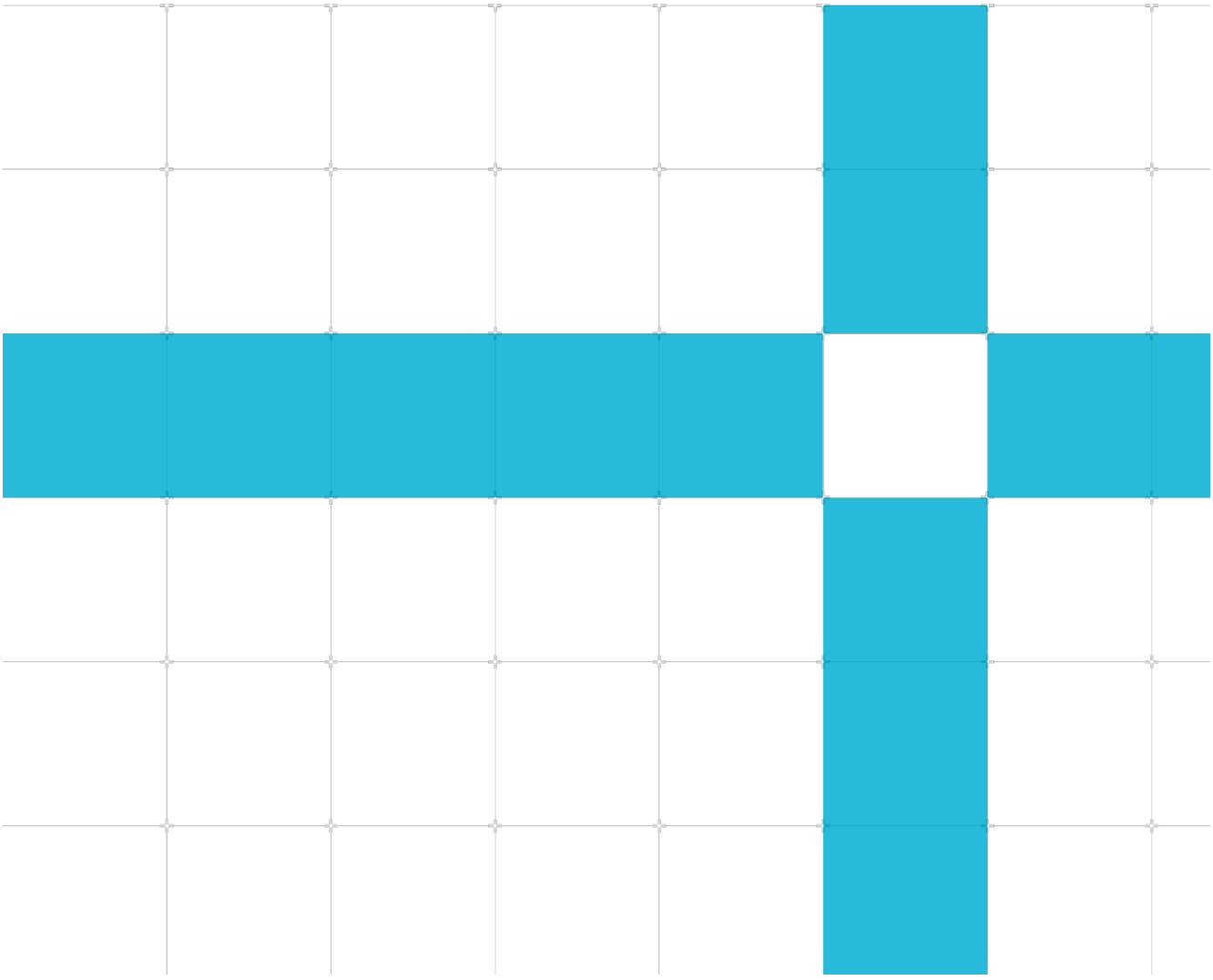
Date of issue: September 10, 2025

Non-Confidential

Document version: 8.0

Document ID: SDEN-3244752

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



This document (SDEN_3244752_8.0_en) was issued on September 10, 2025.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-3244752**

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® C1-Ultra (MP201), 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

r1p0 implementation fixes

Note the following errata might be fixed in some implementations of r1p0. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR_EL1[0]	3705939 RDFFR can return an erroneous value, where all values are set to 1, instead of the correct value set to 0
REVIDR_EL1[1]	3779319 AMEVCNTR14_EL0 and AMEVTCNTR15_EL0 not wired to events Note: This erratum also requires update which sets REVIDR_EL1[5] for erratum 3966815 for full functionality and should only be considered fixed when both REVIDR_EL1[1]=1 and REVIDR_EL1[5]=1
REVIDR_EL1[2]	3782177 Cannot access AMU auxiliary counter enable or disable for auxiliary counters 3, 4, 5
REVIDR_EL1[3]	3865171 Loads issued to Non-Cacheable or Device GRE memory can violate ordering requirements
REVIDR_EL1[5]	3966815 AMU AUX counters 4 and 5 are not counting when PMU is disabled
REVIDR_EL1[6]	3919694 Power transition might deadlock on Utility Bus or APB access

Note that there is no change to the MIDR_EL1 which remains at r1p0 . Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

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.

September 10, 2025: Changes in document version v8.0

ID	Status	Area	Category	Summary
4073280	New	Programmer	Category C	DRPS with invalid SPSR_ELx value incorrectly triggers UNDEF instead of illegal execution state exception
4103522	New	Programmer	Category C	Incorrect implementation of CheckSMEEnabled() function in debug state

August 15, 2025: Changes in document version v7.0

ID	Status	Area	Category	Summary
3865171	New	Programmer	Category B	Loads issued to Non-Cacheable or Device GRE memory can violate ordering requirements
3919694	New	Programmer	Category B	Power transition might deadlock on Utility Bus or APB access
3926381	New	Programmer	Category B	Executing a WFx instruction with PSTATE.SM set might result in processor deadlock
4015814	New	Programmer	Category B	Memory-mapped access might be prohibited when ETADE, EDADE, or EPMADE are set in MDCR_EL3
4043997	New	Programmer	Category B	FEAT_MOPS instructions might cause performance degradation
4095584	New	Programmer	Category B	Incorrect address calculations at 0xFFFF_0000_0000_0000
4102704	New	Programmer	Category B	A load might return incorrect data forwarded from older store
3159181	New	Programmer	Category C	Incorrect count for PMU event 0x002B (L3D_CACHE) might be observed
3745749	New	Programmer	Category C	Update to SCR_EL3.EEL2 might inhibit TLB invalidation
3761991	New	Programmer	Category C	Executing a continuous stream of SME predicated store instrucitons might lead to a denial of service
3830003	New	Programmer	Category C	ISV bit value in the Exception Syndrome Register might be incorrect on a software stepped execution of CLRBHB instruction
3836267	New	Programmer	Category C	SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC PMU events might return incorrect data
3877009	New	Programmer	Category C	Incorrect count for PMU event 0x8285 (L2D_CACHE_PRF) might be observed
3895076	New	Programmer	Category C	PMU event PMU_OVFS is always increased when PMCCNTR_EL0 overflows, independently of PMINTENSET_EL1.C value
3942089	New	Programmer	Category C	LDAPUR, LDAPURB, LDAPURH have stricter memory ordering than required
3963302	New	Programmer	Category C	Incorrect count for PMU event 0x0020 (L2D_CACHE_ALLOCATE) might be observed
3966815	New	Programmer	Category C	AMU AUX counters 4 and 5 are not counting when PMU is disabled
3980765	New	Programmer	Category C	Power off transition might deadlock if debug halt request is asserted
3984952	New	Programmer	Category C	STALL_FRONTEND_FLUSH and CPU_CYCLE related PMU events increment during snoop handling in WFx state
3986295	New	Programmer	Category C	Incorrect exception type reported in ETE exception packet when a SETG* instruction takes a Data Abort exception due to alignment of Xn register
4011447	New	Programmer	Category C	Incorrect count for PMU event 0x81B4 (DSNP_HIT) might be observed
4106082	New	Programmer	Category C	ETE and ELA timestamps are not accurate for core
4128618	New	Programmer	Category C	Incorrect sampling of SPE partial and empty predicate values

January 30, 2025: Changes in document version v6.0

ID	Status	Area	Category	Summary
3651221	New	Programmer	Category B	Unprivileged data memory-dependent prefetches might leak privileged data
3806278	New	Programmer	Category B	Clock is not gated in synchronous configurations when in FULL_RET power mode
3815514	New	Programmer	Category B	Store Allocation Tag (STG) instructions to untagged addresses when in SME streaming mode might cause a deadlock
3727581	New	Programmer	Category C	PMU events based on a governing predicate are counted incorrectly in streaming mode for operations that do not have a vector register as source or destination
3770370	New	Programmer	Category C	Incorrect fault type might be recorded for Data aborts when VTCR_EL2.HAFT is enabled
3770628	New	Programmer	Category C	PMU event 0x3008 reports incorrect counts
3779319	New	Programmer	Category C	AMEVCNTR14_EL0 and AMEVTCNTR15_EL0 not wired to events
3782177	New	Programmer	Category C	Cannot access AMU auxiliary counter enable or disable for auxiliary counters 3, 4, 5
3839259	New	Programmer	Category C	PMU events triggered by predicate operations count incorrectly

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

ID	Status	Area	Category	Summary
3658374	Updated	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS
3684152	New	Programmer	Category B	A PE executing a load instruction that accesses a memory region which crosses a 4K boundary might cause a deadlock
3705939	New	Programmer	Category B	RDFFR can return an erroneous value
3709460	New	Programmer	Category B	Incorrect SME priority computation by the core when SCR_EL3.HXEN = 0
3705500	New	Programmer	Category C	VSET.priority might incorrectly designate non-maskable property (NMI)
3705505	New	Programmer	Category C	ICH_LR<n>_EL2.NMI might incorrectly designate NMI for a Group 0 virtual interrupt
3706468	New	Programmer	Category C	ETE counters failing to count events associated with Streaming Mode Compute Unit (SMCU)
3708289	New	Programmer	Category C	Under-counting PMU events involving Streaming Mode Compute Unit (SMCU)
3709839	New	Programmer	Category C	Incorrect count for PMU event 0x400B (L3D_CACHE_LMISS_RD) might be observed
3716859	New	Programmer	Category C	PMU events SVE_PRED_NOT_FULL_SPEC (evt_0x8079) and SSVE_PRED_NOT_FULL_SPEC (evt_0x3237) count incorrectly
3722310	New	Programmer	Category C	ESR_ELx.ISV and EDSCR.STATUS does not provide load exclusive syndrome information that could be provided for Software Step or Halting Step
3740433	New	Programmer	Category C	A Software Step Exception is possible before the entirety of a non aborting CPY* (Memory Copy) instruction completes
3750110	New	Programmer	Category C	CPU fails to inject an IESB after taking a Data Abort resulting from a FEAT_MOPS instruction
3756951	New	Programmer	Category C	SPE event 22 might be incorrectly filtered or reported as 0 for load that should have set the event to 1

August 30, 2024: Changes in document version v4.0

ID	Status	Area	Category	Summary
3435146	New	Programmer	Category B	Aborted CPU powerdown might lead to incomplete flush of CMC metadata
3627010	New	Programmer	Category B	Read of ICV_RPR_EL1 group priority field might return incorrect data for an interrupt with NMI priority
3658374	New	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS
3683289	New	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock
3460354	New	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception
3517564	New	Programmer	Category C	SPE operation type is corrupted under certain conditions
3613429	New	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability
3628852	New	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled
3631033	New	Programmer	Category C	Lower priority exception might be reported when abort condition is detected at both stages of translation
3653291	New	Programmer	Category C	PE might report an unexpected Tag Check fault on a CPY* instruction accessing tagged memory
3667362	New	Programmer	Category C	FFR might not capture the lowest faulting memory element
3676338	New	Programmer	Category C	PMU events are mis-categorized by not considering the effect of Taken locally
3679665	New	Programmer	Category C	Hang: Test hanging as two LD instructions having the same uid
3699915	New	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors
3701467	New	Programmer	Category C	Parity Error in L2BTB can cause failure in breakpoint detection

June 06, 2024: Changes in document version v3.0

ID	Status	Area	Category	Summary
3502731	New	Programmer	Category B	Loads eligible for memory renaming optimization can violate internal visibility requirement
3435056	New	Programmer	Category C	Read data for ICC_NMIAR1_EL1 might get corrupted by ICV_NMIAR1_EL1 when HCR_EL2.FMO is set
3442673	New	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative
3487900	New	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed
3523918	New	Programmer	Category C	Incorrect decoding of SVE version of PRFH scalar plus vector (gather) instructions

April 18, 2024: Changes in document version v2.0

ID	Status	Area	Category	Summary
3324333	New	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing
2959756	New	Programmer	Category C	LDIAPP instruction to Non-cacheable or Device memory might not enforce order between memory effects associated with its destination registers
3333166	New	Programmer	Category C	PE takes an UNDEFINED exception to the wrong Exception level when accessing TCR2_ELx system registers in Debug state at EL1
3384225	New	Programmer	Category C	PMU event L3D_CACHE_ALLOCATE, L3D_CACHE, and L3D_CACHE_RW count incorrectly
3403076	New	Programmer	Category C	Device-nGRE STILP instructions might not be ordered with regard to younger Device-nGRE stores or atomic instructions

February 29, 2024: 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
3324333	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	r0p0	r1p0
3435146	Programmer	Category B	Aborted CPU powerdown might lead to incomplete flush of CMC metadata	r0p0	r1p0
3502731	Programmer	Category B	Loads eligible for memory renaming optimization can violate internal visibility requirement	r0p0	r1p0
3627010	Programmer	Category B	Read of ICV_RPR_EL1 group priority field might return incorrect data for an interrupt with NMI priority	r0p0	r1p0
3651221	Programmer	Category B	Unprivileged data memory-dependent prefetches might leak privileged data	r0p0	r1p0
3658374	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS	r0p0, r1p0	Open
3683289	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	r0p0	r1p0
3684152	Programmer	Category B	A PE executing a load instruction that accesses a memory region which crosses a 4K boundary might cause a deadlock	r0p0	r1p0
3705939	Programmer	Category B	RDFFR can return an erroneous value	r0p0, r1p0	Open
3709460	Programmer	Category B	Incorrect SME priority computation by the core when SCR_EL3.HXEN = 0	r0p0, r1p0	Open
3806278	Programmer	Category B	Clock is not gated in synchronous configurations when in FULL_RET power mode	r0p0, r1p0	Open
3815514	Programmer	Category B	Store Allocation Tag (STG) instructions to untagged addresses when in SME streaming mode might cause a deadlock	r0p0, r1p0	Open
3865171	Programmer	Category B	Loads issued to Non-Cacheable or Device GRE memory can violate ordering requirements	r0p0, r1p0	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3919694	Programmer	Category B	Power transition might deadlock on Utility Bus or APB access	r0p0, r1p0	Open
3926381	Programmer	Category B	Executing a WFx instruction with PSTATE.SM set might result in processor deadlock	r1p0	Open
4015814	Programmer	Category B	Memory-mapped access might be prohibited when ETADE, EDADE, or EPMADE are set in MDCR_EL3	r0p0, r1p0	Open
4043997	Programmer	Category B	FEAT_MOPS instructions might cause performance degradation	r0p0, r1p0	Open
4095584	Programmer	Category B	Incorrect address calculations at 0xFFFF_0000_0000_0000	r0p0, r1p0	Open
4102704	Programmer	Category B	A load might return incorrect data forwarded from older store	r0p0, r1p0	Open
2959756	Programmer	Category C	LDIAPP instruction to Non-cacheable or Device memory might not enforce order between memory effects associated with its destination registers	r0p0	r1p0
3159181	Programmer	Category C	Incorrect count for PMU event 0x002B (L3D_CACHE) might be observed	r0p0	r1p0
3333166	Programmer	Category C	PE takes an UNDEFINED exception to the wrong Exception level when accessing TCR2_ELx system registers in Debug state at EL1	r0p0	r1p0
3384225	Programmer	Category C	PMU event L3D_CACHE_ALLOCATE, L3D_CACHE, and L3D_CACHE_RW count incorrectly	r0p0	r1p0
3403076	Programmer	Category C	Device-nGRE STILP instructions might not be ordered with regard to younger Device-nGRE stores or atomic instructions	r0p0	r1p0
3435056	Programmer	Category C	Read data for ICC_NMIAR1_EL1 might get corrupted by ICV_NMIAR1_EL1 when HCR_EL2.FMO is set	r0p0	r1p0
3442673	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative	r0p0	r1p0
3460354	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
3487900	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed	r0p0	r1p0
3517564	Programmer	Category C	SPE operation type is corrupted under certain conditions	r0p0	r1p0
3523918	Programmer	Category C	Incorrect decoding of SVE version of PRFH scalar plus vector (gather) instructions	r0p0	r1p0
3613429	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability	r0p0	r1p0
3628852	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled	r0p0	r1p0
3631033	Programmer	Category C	Lower priority exception might be reported when abort condition is detected at both stages of translation	r0p0	r1p0
3653291	Programmer	Category C	PE might report an unexpected Tag Check fault on a CPY* instruction accessing tagged memory	r0p0	r1p0
3667362	Programmer	Category C	FFR might not capture the lowest faulting memory element	r0p0	r1p0
3676338	Programmer	Category C	PMU events are mis-categorized by not considering the effect of Taken locally	r0p0	r1p0
3679665	Programmer	Category C	Hang: Test hanging as two LD instructions having the same uid	r0p0, r1p0	Open
3699915	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors	r0p0, r1p0	Open
3701467	Programmer	Category C	Parity Error in L2BTB can cause failure in breakpoint detection	r0p0, r1p0	Open
3705500	Programmer	Category C	VSET.priority might incorrectly designate non-maskable property (NMI)	r0p0, r1p0	Open
3705505	Programmer	Category C	ICH_LR<n>_EL2.NMI might incorrectly designate NMI for a Group 0 virtual interrupt	r0p0, r1p0	Open
3706468	Programmer	Category C	ETE counters failing to count events associated with Streaming Mode Compute Unit (SMCU)	r0p0, r1p0	Open
3708289	Programmer	Category C	Under-counting PMU events involving Streaming Mode Compute Unit (SMCU)	r0p0, r1p0	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3709839	Programmer	Category C	Incorrect count for PMU event 0x400B (L3D_CACHE_LMISS_RD) might be observed	r0p0, r1p0	Open
3716859	Programmer	Category C	PMU events SVE_PRED_NOT_FULL_SPEC (evt_0x8079) and SSVE_PRED_NOT_FULL_SPEC (evt_0x3237) count incorrectly	r0p0, r0p1	Open
3722310	Programmer	Category C	ESR_ELx.ISV and EDSCR.STATUS does not provide load exclusive syndrome information that could be provided for Software Step or Halting Step	r0p0, r1p0	Open
3727581	Programmer	Category C	PMU events based on a governing predicate are counted incorrectly in streaming mode for operations that do not have a vector register as source or destination	r0p0, r1p0	Open
3740433	Programmer	Category C	A Software Step Exception is possible before the entirety of a non aborting CPY* (Memory Copy) instruction completes	r0p0, r1p0	Open
3745749	Programmer	Category C	Update to SCR_EL3.EEL2 might inhibit TLB invalidation	r0p0, r1p0	Open
3750110	Programmer	Category C	CPU fails to inject an IESB after taking a Data Abort resulting from a FEAT_MOPS instruction	r0p0, r1p0	Open
3756951	Programmer	Category C	SPE event 22 might be incorrectly filtered or reported as 0 for load that should have set the event to 1	r0p0, r1p0	Open
3761991	Programmer	Category C	Executing a continuous stream of SME predicated store instrucitons might lead to a denial of service	r0p0, r1p0	Open
3770370	Programmer	Category C	Incorrect fault type might be recorded for Data aborts when VTCR_EL2.HAFT is enabled	r0p0, r1p0	Open
3770628	Programmer	Category C	PMU event 0x3008 reports incorrect counts	r0p0, r1p0	Open
3779319	Programmer	Category C	AMEVCNTR14_EL0 and AMEVTCNTR15_EL0 not wired to events	r0p0, r1p0	Open
3782177	Programmer	Category C	Cannot access AMU auxiliary counter enable or disable for auxiliary counters 3, 4, 5	r0p0, r1p0	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3830003	Programmer	Category C	ISV bit value in the Exception Syndrome Register might be incorrect on a software stepped execution of CLRBHB instruction	r0p0, r1p0	Open
3836267	Programmer	Category C	SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC PMU events might return incorrect data	r0p0, r1p0	Open
3839259	Programmer	Category C	PMU events triggered by predicate operations count incorrectly	r0p0, r1p0	Open
3877009	Programmer	Category C	Incorrect count for PMU event 0x8285 (L2D_CACHE_PRF) might be observed	r0p0, r1p0	Open
3895076	Programmer	Category C	PMU event PMU_OVFS is always increased when PMCCNTR_EL0 overflows, independently of PMINTENSET_EL1.C value	r0p0, r1p0	Open
3942089	Programmer	Category C	LDAPUR, LDAPURB, LDAPURH have stricter memory ordering than required	r0p0, r1p0	Open
3963302	Programmer	Category C	Incorrect count for PMU event 0x0020 (L2D_CACHE_ALLOCATE) might be observed	r0p0, r1p0	Open
3966815	Programmer	Category C	AMU AUX counters 4 and 5 are not counting when PMU is disabled	r0p0, r1p0	Open
3980765	Programmer	Category C	Power off transition might deadlock if debug halt request is asserted	r0p0, r1p0	Open
3984952	Programmer	Category C	STALL_FRONTEND_FLUSH and CPU_CYCLE related PMU events increment during snoop handling in WFx state	r0p0, r1p0	Open
3986295	Programmer	Category C	Incorrect exception type reported in ETE exception packet when a SETG* instruction takes a Data Abort exception due to alignment of Xn register	r0p0, r1p0	Open
4011447	Programmer	Category C	Incorrect count for PMU event 0x81B4 (DSNP_HIT) might be observed	r0p0, r1p0	Open
4073280	Programmer	Category C	DRPS with invalid SPSR_ELx value incorrectly triggers UNDEF instead of illegal execution state exception	r0p0, r1p0	Open
4103522	Programmer	Category C	Incorrect implementation of CheckSMEEnabled() function in debug state	r1p0	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
4106082	Programmer	Category C	ETE and ELA timestamps are not accurate for core	r0p0, r1p0	Open
4128618	Programmer	Category C	Incorrect sampling of SPE partial and empty predicate values	r0p0, r1p0	Open

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

3324333
MSR PSTATE.SSBS to 0 is not fully self-synchronizing

Status

Fault type: Programmer Category B
Fault status: Present in r0p0. Fixed in r1p0.

Description

When PSTATE.SSBS is written to 0, the Arm Architecture specifies that side-effects are guaranteed to be visible to later instructions in the Execution stream. However, for a window of time during speculative execution of MSR PSTATE.SSBS, speculative store data bypassing might still occur.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following condition applies:

MSR PSTATE.SSBS executes, setting PSTATE.SSBS to 0.

Implications

Security sensitive code executed shortly after MSR PSTATE.SSBS to 0 might not be fully protected by the *Speculative Store Bypass Safe* (SSBS) feature.

Workaround

Software at EL3, EL2, and EL1 should follow writes to the SSBS register with a *Speculation Barrier* (SB) instruction to ensure that the new value of PSTATE.SSBS affects subsequent instructions in the execution stream under speculation.

A kernel at EL1 or EL2 should not advertise the presence of MRS/MSR instructions to read/write the SSBS register from EL0. Arm expects that kernels provide system calls for EL0 software to modify PSTATE.SSBS when the SSBS register is not implemented and that EL0 software will use this when the presence of the SSBS register is not advertised.

3435146
Aborted CPU powerdown might lead to incomplete flush of CMC metadata

Status

Fault type: Programmer Category B
Fault status: Present in r0p0. Fixed in r1p0.

Description

Correlated Miss Cache (CMC) prefetcher metadata is managed by hardware and flushed on a context switch to avoid prefetcher training information from one context to be used by another. A PE in the process of flushing CMC metadata when a powerdown request arrives will stop the CMC flush and proceed with the powerdown sequence instead. The powerdown sequence is terminated by the detection of a RAS error in the PE, or the assertion of the PABANDON input to the PE.

Configurations affected

This erratum affects all configurations with CMC prefetching enabled through CPUECTLR_EL1[63:61].

Conditions

The erratum occurs if all the following conditions apply:

1. PE has CMC prefetching enabled
2. A CMC metadata flush is triggered by a context switch or CTX instruction
3. PE initates a powerdown sequence which includes a WFI instruction
4. At least one of the following conditions apply:
    - PE detects a RAS error and signals a Fault Handling Interrupt or Error Recovery Interrupt
    - SME engine denies the powerdown request due to PSTATE.SM or ZA being set
    - PE detects assertion of PABANDON input

Implications

When the previous conditions are met, the PE will remain in the RUN state, but the CMC metadata flush will be incomplete. This might lead to prefetches in the current context being issued based on training information from a different context. Note that CMC prefetching is Virtual Address based and therefore subject to successful MMU translation in the current context.

Workaround

This erratum can be avoided by explicitely flushing the CMC metadata as part of the powerdown sequence. This is accomplished by replacing the WFI instruction in the powerdown sequence with the following instructions sequence:

CPP RCTX
DSB
WFI

3502731
Loads eligible for memory renaming optimization can violate internal visibility requirement

Status

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

Description

A load that is eligible for memory renaming optimization can violate internal visibility requirement.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following sequence of conditions apply:

1. A sequence of instructions is executed, potentially with intervening instructions:
   a. A store that is not an exclusive.
   b. A load that is not an exclusive which loads bytes overlapping with the store.
   c. A load that is not an exclusive and gets all of its data from the store.
2. Another observer performs a store to one or more of the overlapping bytes which is coherence-after the store.
3. Unusual micro-architectural conditions occur.

Implications

When the above conditions are met, the earlier load might read newer data than the later load, violating internal visibility requirement.

Workaround

The erratum can be avoided by setting CPUACTLR4[23] to 1'b1 which will disable Memory Renaming optimization. The performance impact of setting this chicken bit is about 0.82% in GB6.

3627010
Read of ICV_RPR_EL1 group priority field might return incorrect data for an interrupt with NMI priority

Status

Fault type: Programmer Category B
Fault status: Present in r0p0. Fixed in r1p0.

Description

When ICV_RPR_EL1 is read when a virtual interrupt with *Non-Maskable Interrupt* (NMI) priority is the highest priority active virtual interrupt, the group priority bits should read as zeroes (and bit 63 should be read as 1 as this is the NMI designation bit). If 2 virtual interrupts are active simultaneously (1 NMI and 1 non-NMI), the group priority field of the ICV_RPR_EL1 will return the value of the non-NMI even though the NMI is assumed to be the highest priority active virtual interrupt.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The PE is executing at EL1
2. Common GIC registers are virtualized (HCR_EL2.IMO == 1 OR HCR_EL2.FMO == 1)
3. The PE has 2 simultaneously active virtual interrupts (1 with NMI priority, 1 without NMI priority)
4. The PE executes an MRS <dst>, ICC_RPR_EL1 instruction

Implications

If the conditions are met, the MRS <dst>, ICC_RPR_EL1 instruction will erroneously return the group priority of the non-NMI instruction while the architecture requires that the group priority field should read as zeroes when the highest priority active virtual interrupt has NMI priority. The effects should be negligible as software should generally not care about the group priority bits if the NMI bit is set.

Workaround

The workaround is to look at the NMI bit first when the ICC_RPR_EL1 is read for an NMI, and then subsequently ignore the group priority bits.

3651221
Unprivileged data memory-dependent prefetches might leak privileged data

Status

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

Description

An unprivileged context can trigger a data memory-dependent prefetch engine to fetch the contents of a privileged location for which it does not have read permission, and consume those contents as an address that is also dereferenced.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

- Data memory-dependent prefetch engine is trained on an adversary's data and then deployed to access a privileged location.

Implications

An unprivileged (EL0) attacker can force the prefetcher to load content from a privileged location in the same translation regime and dereference it despite the permission checks and TCR.E0PDx. This secret-dependent access might leave a trace in data caches and TLBs that could be measured by the attacker to recover the secret.

This issue does not affect guest-to-guest and guest-to-hypervisor isolation guarantees. Likewise, in configurations with RME enabled, *Granule Protection Checks* (GPC) are honoured by the prefetcher.

Workaround

The erratum can be avoided by disabling the affected prefetcher setting CPUACTLR6_EL1 [41].

3658374
Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS

Status

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

Description

When ICH_VMCR_EL2.VBPR1 is written in Secure state (SCR_EL3.NS==0) and then subsequently read in Non-secure state (SCR_EL3.NS==1), a wrong value might be returned. The same issue exists in the opposite way: write in Non-secure state and read in Secure state. ICH_VMCR_EL2.VBPR1 is an alias of ICV_BPR1_EL1 which is architecturally defined as NOT banked. The RTL erroneously has this register implemented as two separate registers (secure and non-secure copies) banked by SCR_EL3.NS.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs if all the following conditions apply:

1. The *Processing Element* (PE) is executing at EL3
2. SCR_EL3.NS == 1 or 0
3. The PE executes an MSR ICH_VMCR_EL2.VBPR1 instruction
4. SCR_EL3.NS == 0 or 1 (the opposite value from when the MSR occurred)
5. The PE executes an MRS <dst>, ICH_VMCR_EL2.VBPR1 instruction

Implications

If the previous conditions are met, the MRS <dst>, ICH_VMCR_EL2.VBPR1 instruction will erroneously return the value that was last written to this field with the opposite SCR_EL.NS value from which it was read (or the reset value if it was never written in that security state).

Workaround

The workaround is for EL3 software that performs context save/restore on a change of Security state to use a value of SCR_EL3.NS when accessing ICH_VMCR_EL2 that reflects the Security state that owns the data being saved or restored. For example, EL3 software should set SCR_EL3.NS to 1 when saving or restoring the value ICH_VMCR_EL2 for Non-secure (or Realm) state. EL3 software should clear SCR_EL3.NS to 0 when saving or restoring the value ICH_VMCR_EL2 for Secure state.
Further, when ICV_CTLR_EL1.CBPR is set to 1, software reads of ICV_BPR1_EL1 at Secure EL1 and Secure EL2 will incorrectly return (ICV_BPR0_EL1 +1) saturated to b1111, instead of the unmodified value. of ICV_BPR0_EL1. This erratum does not affect the presentation of virtual interrupts as exceptions, or the result of reads of the ICV_IARx_EL1 registers.

3683289
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 r1p0.

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.

3684152
A PE executing a load instruction that accesses a memory region which crosses a 4K boundary might cause a deadlock

Status

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

Description

Under certain unusual micro-architectural conditions, a *Processing Element* (PE) executing a load instruction that accesses a memory region which crosses a 4K boundary might deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. PE executes a load instruction that accesses a memory region crossing a 4K boundary.
2. Unusual micro-architectural conditions occur.

Implications

When the above conditions are met, the PE might deadlock.

Workaround

The erratum can be avoided by setting CPUACTLR_EL1[60:58] == 3'b001, which has a small perf impact.

3705939
RDFFR can return an erroneous value

Status

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

Description

RDFFR can return an erroneous value, where all values are set to 1, instead of the correct value set to 0.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum can occur when executing code that involves this sequence of instructions, possibly mixed with other instructions, but in this specified order:

1. SETFFR
2. SMSTART
3. SMSTOP
4. RDFFR

Implications

If the previous conditions are met, under certain microarchitectural conditions the RDFFR instruction might erroneously return all values set to 1, when all the correct values should be 0 for the FFR.

Workaround

The erratum can be avoided by setting CPUACTLR_EL1[48] to 1'b1 which disables a RDFFR optimization. Setting this bit has negligible impact on GB6/SPECint performance, but will have an impact on SVE RDFFR performance. Please contact Arm for more details.

3709460
Incorrect SME priority computation by the core when SCR_EL3.HXEN = 0

Status

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

Description

The core might execute Streaming SVE execution with the wrong Streaming Mode priority.

Configurations affected

This erratum affects all configurations.

Conditions

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

- SCR_EL3.HXEN is programmed to 0, or HCR_EL2.{E2H,TGE} are {1,1}.
- EL2 is Enabled and PSTATE.EL = 0 or 1.
- HCRX_EL2.SMPME is programmed to 1.

Implications

If the previous conditions are met, the effective Streaming execution priority is the result of the remapping of SMPRI_EL1 priority by SMPRIMAP_EL2, while this remapping should have been deactivated.

Workaround

When exiting a guest to run host software, a type-2 hypervisor must set SMPRIMAP_EL2 to provide a 1:1 mapping of priorities, because this register is not out-of-context when HCR_EL2.{E2H,TGE} is {1,1}.

3806278
Clock is not gated in synchronous configurations when in FULL_RET power mode

Status

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

Description

In synchronous configurations, the clock is not gated correctly when the core is in FULL_RET power mode. This can cause corruption of the retention state when FULL_RET power mode is implemented with some retention technologies.

Configurations affected

This erratum affects configurations where the CORE<n>_ASYNC_BRIDGE parameter is set to FALSE.

Conditions

The erratum occurs if the following sequence of conditions apply:

1. The IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL or IMP_CPUPWRCTLR_EL1.WFE_RET_CTRL fields are set to a non-zero value to enable FULL_RET power mode.
2. The core executes Wait for Interrupt (WFI) or Wait for Event (WFE) and remains idle for long enough to enter FULL_RET power mode.

Implications

When the core enters FULL_RET power mode, the clock will not be gated which can cause electrical issues with some retention technologies. This might lead to corruption of the retained state.

Workaround

Systems that implement FULL_RET power mode with a retention technology that requires the clock to be gated must disable the use of FULL_RET power mode. This can be done by setting the IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL and IMP_CPUPWRCTLR_EL1.WFE_RET_CTRL fields to 0 which is their default value after reset.

3815514
Store Allocation Tag (STG) instructions to untagged addresses when in SME streaming mode might cause a deadlock

Status

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

Description

Under certain rare microarchitectural conditions, executing two or more *Store Allocation Tag* (STG) instructions to untagged addresses when the core is in *Scalable Matrix Extension* (SME) streaming mode might cause a deadlock.

Configurations Affected

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

Conditions

The erratum occurs if all the following conditions apply:

- *Processing Element* (PE) is configured with ERR0CTLR.ED = 1, enabling Error detection and correction.
- The core is in SME streaming mode (PSTATE.SM = 1)
- Two or more cacheable STG instructions, that neither zero nor store data, are executed in close proximity to the same cache line but different 32-byte locations.
- The STG instructions are to untagged addresses.
- An error response is received for a read to the untagged address (after the core is in SME streaming mode).
- Other micro-architectural conditions occur.

Implications

When the previous conditions are met, then the STG instructions might never complete.

Workaround

This erratum can be avoided by setting CPUACTLR5_EL1[13] to 1.

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

3865171
Loads issued to Non-Cacheable or Device GRE memory can violate ordering requirements

Status

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

Description

Under certain conditions, a load issued to Non-Cacheable or Device GRE memory can read stale data brought in by an earlier load to the same cache-line thereby violating ordering requirements.

Configurations affected

This erratum affects all configurations.

Conditions

A sequence of instructions is executed, potentially with intervening instructions:

1. A load executes to Non-Cacheable or Device GRE cache-line address A.
2. An instruction executes that creates any of the following order with a younger load instruction:
    - A dependency-ordered-before through address dependency involving update to page tables
    - Fetch-ordered-before by modification to instructions executed
    - Barrier-ordered-before using a load acquire with the younger load being a non-temporal load.
3. Younger Load to Non-Cacheable or Device GRE cache-line address A.
4. Specific micro-architectural timing conditions occur.

Implications

When the previous conditions are met, the younger load in condition 3 might read stale data brought in by the load in condition 1, thereby violating address dependency ordering or fetch-ordered-before requirement or barrier-ordered-before requirement listed in condition 2.

Workaround

The erratum can be avoided by setting CPUACTLR2[22] to 1'b1 which will disable linking multiple Non-Cacheable or Device GRE loads to the same read request for the cache-line. This might have a significant performance impact to Non-cacheable and Device GRE read bandwidth for streaming scenarios.

3919694
Power transition might deadlock on Utility Bus or APB access

Status

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

Conditions

The erratum occurs under the following conditions:

1. The core power transitions from ON to OFF, or OFF_EMU to OFF.
2. One of the following three sequences occurs:
    - i. The core starts a power transition from OFF to ON soon after the previous transition has completed.
      ii. An access is made on the Utility Bus or debug APB interface to access a memory mapped register in the core during the OFF to ON 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.
    - 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. 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.
    - 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.

Implications

If the previous conditions occur, the core might not complete the power transition or the Utility Bus or debug APB access, 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 the power 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.

3926381
Executing a WFx instruction with PSTATE.SM set might result in processor deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r1p0. Open.

Description

A WFx instruction executed while PSTATE.SM is set and the processor is attempting to gain arbitration to CME should result in a DISCONNECT_REQ packet delivered to CME prior to entering a low-power state.

Under rare instruction sequence and internal timing circumstances, the processor might fail to send a DISCONNECT_REQ packet and thus deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following sequence of conditions apply:

1. A significant number (90+) of Fast Context Switch opcodes are executed (processor is accessing CME context (Z, ZA, or ZT0 registers) while not arbitrated to the CME).
2. A WFx instruction is executed while PSTATE.SM is set.
3. The first instruction after the WFx is an SME, SME2, or SVE instruction that requires CME arbitration to execute.
4. Any of the following conditions apply, with additional internal timing requirements while the WFx instruction is executing:
    - Memory-mapped registers in external debug interface are written to disable Embedded Trace, taking ETE from enabled to disabled state
    - Memory-mapped registers in external debug interface are written to disable Branch Record Buffer tracing, taking BRBE from enabled to disabled state
    - Statistical Profile Extension is enabled and the WFx instruction is randomly selected as the sampled operation

Implications

The processor might deadlock.

Workaround

WFx is typically not executed when PSTATE.SM=1 or PSTATE.ZA=1, therefore Arm expects occurrences of this erratum will be unlikely.

The erratum can be avoided by converting WFx and WFxT instructions to NOP when PSTATE.SM=1. This workaround is implemented by executing the following code sequence at EL3, as soon as possible after boot. After it is applied, the code only converts WFx and WFxT instructions to NOP when PSTATE.SM=1 or when PSTATE.ZA=1.

```
// Convert WFx to NOP
   LDR x0,=0x0
   MSR S3_6_c15_c8_0,x0 // MSR CPUPSELR_EL3, X0
   LDR x0,=0xD503205f
   MSR S3_6_c15_c8_2,x0 // MSR CPUPOR_EL3, X0
   LDR x0,=0xFFFFFFDF
   MSR S3_6_c15_c8_3,x0 // MSR CPUPMR_EL3, X0
   LDR x0,=0x1000002043ff
   MSR S3_6_c15_c8_1,x0 // MSR CPUPCR_EL3, X0

// Convert WFxT to NOP
   LDR x0,=0x1
   MSR S3_6_c15_c8_0,x0 // MSR CPUPSELR_EL3, X0
   LDR x0,=0xD5031000
   MSR S3_6_c15_c8_2,x0 // MSR CPUPOR_EL3, X0
   LDR x0,=0xFFFFFFC0
   MSR S3_6_c15_c8_3,x0 // MSR CPUPMR_EL3, X0
   LDR x0,=0x1000002043ff
   MSR S3_6_c15_c8_1,x0 // MSR CPUPCR_EL3, X0
   ISB
```

4015814
Memory-mapped access might be prohibited when ETADE, EDADE, or EPMADE are set in MDCR_EL3

Status

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

Description

The Processing Element (PE) might incorrectly prohibit access to externally accessible registers on the memory-mapped interface based on programming of MDCR_EL3.

Configurations affected

This erratum affects configurations where FEAT_RME is not implemented or FEAT_RME is not enabled.

Conditions

The erratum occurs under the following conditions:

1. A write sets MDCR_EL3 bit fields ETADE, EDADE, or EPMADE
2. A debugger attempts to access registers which are made inaccessible by setting these bit fields

Implications

Under these conditions, an external debugger is unable to access the affected external registers.

Workaround

Software should not write to ETADE, EDADE, or EPMADE access control bit fields of MDCR_EL3 for configurations where FEAT_RME is not implemented or FEAT_RME is not enabled and memory-mapped access to the registers controlled by these fields is desired.

4043997
FEAT_MOPS instructions might cause performance degradation

Status

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

Description

Under certain micro-architectural conditions, when the Processing Element (PE) is executing FEAT_MOPS instructions, performance might be degraded. This is due to micro-architectural flushes that occur due to read-after-write hazards or hardware prefetch ineffectively caching contiguous accesses.

Configurations Affected

All configurations

Conditions

This erratum occurs when all the following conditions apply:

1. FEAT_MOPS instruction is executed
2. At least one of the following conditions are true:
    a. Memory access instruction is executed with addresses near the range of the FEAT_MOPS instruction
    b. Hardware prefetcher is unable to cache contiguous blocks of memory

Implications

When the above conditions are met, the PE might experience performance degradation compared to code compiled without FEAT_MOPS for the same functionality, such as SIMD implementation.

Workaround

This erratum can be avoided by the OS disabling FEAT_MOPS or using alternative memory copy/move sequences that do not rely on FEAT_MOPS.

4095584
Incorrect address calculations at 0xFFFF_0000_0000_0000

Status

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

Description

When the PE takes an exception, a branch, or a Statistical Profiling Extension (SPE) event from PC 0xFFFF_0000_0000_0000, the expected ELR_ELx, trace, SPE, or Branch Record Buffer Extension (BRBE) address value would be 0xFFFF_0000_0000_0000 or 0xFFFF_0000_0000_0004 depending on the cause.
When the erratum happens, the ELR_ELx, trace, SPE, or BRBE address is instead updated with the value 0x0001_0000_0000_0000 or 0x0001_0000_0000_0004 depending on the cause.

Configurations affected

All configurations.

Conditions

The erratum occurs when all the following conditions are met:

- The PE is using the EL0&1 or EL0&2 translation regime.
- An exception, a flush, or a branch is taken from PC 0xFFFF_0000_0000_0000.
- Execution at this PC occurs due to one of the following events:
  - An ESB instruction that synchronizes a pending SError
  - An architectural exception taken at this address, which triggers an IESB that synchronizes a pending SError
  - An architectural exception or exit from Debug State taken to this address
  - A micro-architectural flush that might arise from any of the following:
    - A load or store instruction
    - A change to the instruction opcode at 0xFFFF_0000_0000_0000 since the last execution at the same PC
    - micro-architectural synchronization

Implications

If the above conditions are met, the ELR_ELx, trace, SPE, or BRBE address will be set to 0x0001_0000_0000_0000 or 0x0001_0000_0000_0004 (depending on the cause) rather than the expected value of 0xFFFF_0000_0000_0000 or 0xFFFF_0000_0000_0004.

In the case of repeated micro-architectural flushes this may result in spurious Instruction Aborts.

Executing an ERET instruction with ELR_ELx set to this incorrect value will trigger an Instruction Abort due to the top bits of the Virtual Address (VA) being non-canonical.

Workaround

If the software avoids execution at address 0xFFFF_0000_0000_0000, the erratum cannot occur.

4102704
A load might return incorrect data forwarded from older store

Status

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

Description

When there are multiple stores to the same *Physical Address* (PA), under certain conditions, a load to the same PA might end up forwarding stale data from the wrong store. This might result in the incorrect data being propagated to the load and its consumers.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following sequence of conditions apply:

1. CPUACTLR4_EL1[23] is 0 (this is the reset value).
2. A store is executed which writes to *Physical Address* (PA1).
3. A mixture of stores of GPRs and vector registers are executed which write to arbitrary physical addresses.
4. Complex microarchitecture conditions occur.
5. A load is executed which reads from PA1.

Implications

When the above sequence of conditions are met, the load in step 5 might return stale data for PA1 that is older than the store in step 2.

Workaround

This erratum can be avoided by setting CPUACTLR4_EL1[23] to 1. Overall expected performance degradation is ~1.36%, but isolated benchmark components might see higher or lower impact.

Category B (rare)

There are no errata in this category.

Category C

2959756
LDIAPP instruction to Non-cacheable or Device memory might not enforce order between memory effects associated with its destination registers

Status

Fault type: Programmer Category C
Fault status: Present in r0p0. Fixed in r1p0.

Description

A cache-line crossing LDIAPP instruction to Non-cacheable or Device memory might not enforce order between memory effects associated with its destination registers.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following sequence of conditions apply:

1. PE executes a cache-line crossing LDIAPP instruction to either Non-cacheable or Device memory.
2. Memory access associated with each destination register is aligned to the element size.

Implications

When the previous conditions are met, memory effects associated with Xt1/Wt1 might not be ordered before memory effects associated with Xt2/Wt2.

Workaround

Arm expects that LDIAPP instructions are not commonly used for Non-cacheable or device memory. If a workaround is required, then please contact Arm.

3159181
Incorrect count for PMU event 0x002B (L3D_CACHE) might be observed

Status

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

Description

An L2 cache copyback that results in an L3 cache access might not be counted correctly in PMU event 0x002B (L3D_CACHE).

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. PMU counters are configured to count event 0x002B.
2. L2 cache copyback transaction occurs.

Implications

If the previous conditions are met, the count indicated by event 0x002B (L3D_CACHE) will not correctly reflect the L3 cache accesses initiated by the L2 cache copyback transactions.

Workaround

There is no workaround.

3333166
PE takes an UNDEFINED exception to the wrong Exception level when accessing TCR2_ELx system registers in Debug state at EL1

Status

Fault type: Programmer Category C
Fault status: Present in r0p0. Fixed in r1p0.

Description

When executing in Debug state at EL1 and the PE accesses a TCR2_ELx system register by executing an MRS/MSR instruction, an UNDEFINED exception will occur to the wrong target Exception level. The target Exception level should be EL1, but PE will go to EL2.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under all of the following conditions:

1. PE is in Debug state at Exception level EL1
2. An MSR/MRS instruction accesses TCR2_ELx system registers
3. SCR_EL3.TCR2En = 0
4. EDSCR.SDD = 1

Implications

If all the previous conditions are met, the MRS/MSR instruction will cause an UNDEFINED Exception to EL2, when the correct target Exception level should be EL1.

Workaround

This erratum has no workaround.

3384225
PMU event L3D_CACHE_ALLOCATE, L3D_CACHE, and L3D_CACHE_RW count incorrectly

Status

Fault type: Programmer Category C
Fault status: Present in r0p0. Fixed in r1p0.

Description

Counting of the L3D_CACHE_ALLOCATE, L3D_CACHE, and L3D_CACHE_RW PMU events are incorrect.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs whenever the L3D_CACHE_ALLOCATE, L3D_CACHE, or L3D_CACHE_RW PMU events (numbers 0x29, 0x2b, and 0x8150 respectively) are selected for counting.

Implications

The PMU events L3D_CACHE_ALLOCATE, L3D_CACHE, and L3D_CACHE_RW count unpredictably and cannot be used for useful analysis.

Workaround

This erratum has no workaround.

3403076
Device-nGRE STILP instructions might not be ordered with regard to younger Device-nGRE stores or atomic instructions

Status

Fault type: Programmer Category C
Fault status: Present in r0p0. Fixed in r1p0.

Description

A Device-nGRE STILP instruction that is not aligned to the total size might not be ordered before younger nGRE stores or atomic instructions which have an address overlap.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following sequence of conditions apply:

1. A PE executes a Device-nGRE STILP instruction that is not aligned to the total size.
2. A younger nGRE store or atomic instruction which has an address overlap with the older STILP instruction is executed.

Implications

If the previous conditions are met, then the memory accesses might not be performed in the correct order, as determined by the sequential execution model.

Workaround

Arm expects that STILP instructions are not commonly used for device memory. If a workaround is required, then please contact Arm.

3435056
Read data for ICC_NMIAR1_EL1 might get corrupted by ICV_NMIAR1_EL1 when HCR_EL2.FMO is set

Status

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

Description

When HCR_EL2.FMO is on and HCR_EL2.IMO is off and there's a pending physical IRQ with non-maskable-interrupt (NMI) priority, a read of the ICC_NMIAR1_EL1 register will incorrectly return special interrupt number 1023.

Configurations Affected

This erratum affects all configurations.

Conditions:

This erratum occurs under the following conditions:

1. The PE is executing at EL1
2. EL2 is enabled in the current security state
3. HCR_EL2.{IMO,FMO} == {0,1}
4. SCTLR_EL1.NMI == 1
5. The PE has pending physical IRQ with non-maskable interrupt (NMI) priority
6. The PE executes an **`MRS <dst>, ICC_NMIAR1_EL1`** instruction

Implications

If the conditions are met, the **`MRS <dst>, ICC_NMIAR1_EL1`** instruction will acknowledge and activate the pending IRQ with NMI priority, however the value returned in the **`<dst>`** register will be the special interrupt ID 1023 rather than the interrupt ID of the IRQ.

This might lead to deadlock as software will not be aware that the IRQ had been acknowledged and will not know to deactivate the IRQ. The active priority might prevent the delivery and acknowledgement of other interrupts.

Workaround

No workaround is expected to be required. Arm expects that software will configure HCR_EL1.{IMO,FMO} as either {1,1} or {0,0} whenever EL1 can execute an **`MRS <dst>, ICC_NMIAR1_EL1`** instruction.

3442673
PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative

Status

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

Description

When software directly writes PSTATE.PAN or PSTATE.UAO with an MSR instruction, the Arm Architecture specifies that side-effects are guaranteed to be visible to later instructions in the Execution stream. However, for a window of time prior to the execution of MSR PSTATE.{PAN,UAO}, instructions following the MSR might speculatively execute with the old context, prior to re-executing non-speculatively under the new, expected context.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following condition applies:

- MSR PSTATE.{PAN or UAO} executes

Implications

Speculative execution of instructions using stale PSTATE.{UAO,PAN} context could in theory present a window of opportunity for a security attack. However, Arm security team has evaluated the practical risk to be very low, given the use-cases of the bits in question and the complexity involved in exploiting.

Workaround

A workaround is not expected to be required.

3460354
EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception

Status

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

Description

When a Load-Exclusive instruction is executed with Halting Step enabled, EDSCR.STATUS is not updated if the Load-Exclusive instruction causes a synchronous exception.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. In Debug state, the debugger enables Halting Step
2. Debug state is exited and a Load-Exclusive instruction (LDX*/LDAX*) is stepped
3. The Load-Exclusive generates a synchronous exception while executing

Implications

If the conditions are met, EDSCR.STATUS will not be updated.

Workaround

There is no workaround.

3487900
Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed

Status

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

Description

A hardware generated prefetch operation or a PRFM instruction might indicate a L1D_TLB_REFILL_RD event leading to an incorrect count.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. PMU counters are configured to count event 0x004C.
2. A hardware generated prefetch or PRFM instruction might encounter a L1D TLB miss, resulting in a refill operation and triggering event 0x004C.

Implications

If the previous conditions are met, the count indicated by event 0x004C will not reflect the conditions specified in the Arm Architecture Reference Manual. Furthermore, this event is used in calculating the "Attributable Level 1 TLB refill rate, read" metric which by extension will not reflect an accurate rate.

Workaround

No workaround is required unless PMU event 0x004C is required. If a workaround is needed, this erratum can be avoided by counting three separate PMU events in place of event 0x004C:

- Event 0x0005 (L1D_TLB_REFILL)
- Event 0x004D (L1D_TLB_REFILL_WR)
- Event 0x10E. (L1D_TLB_REFILL_RD_PF)

These events can be used to calculate an Effective event 0x004C as follows:
Effective Event 0x004C = Event 0x0005 - Event 0x004D - Event 0x010E

Effective event 0x004C can be used in place of event 0x004C in calculation of "Attributable Level 1 TLB refill rate, read" to provide an accurate rate calculation.

Arm Architecture Reference Manual relevant events:

Mnemonic	Number
L1D_TLB_REFILL	0x0005
L1D_TLB_REFILL_RD	0x004C
L1D_TLB_REFILL_WR	0x004D
L1D_TLB_RD	0x004E

Implementation Defined relevant event:

Mnemonic	Number
L1D_TLB_REFILL_RD_PF	0x010E

Arm Architecture Reference Manual relevant metric:
"Attributable Level 1 TLB refill rate, read" (Event 0x004C / Event 0x004E)

3517564
SPE operation type is corrupted under certain conditions

Status

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

Description

The FP field (Floating Point) of the operation type header in a *Statistical Profiling Extension* (SPE) record, might not be set correctly for certain *Scalable Vector Extension* (SVE) samples. The affected opcodes are FDIV, FDIVR and FSQRT.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. SPE sampling is enabled.
2. SPE samples one of the following instructions:
    - FDIV
    - FDIVR
    - FSQRT

Implications

If the previous conditions are met, then the FP bit information in the SPE buffer might be inaccurate for the previous mentioned samples.

Workaround

There is no workaround.

3523918
Incorrect decoding of SVE version of PRFH scalar plus vector (gather) instructions

Status

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

Description

The 32-bit unpacked scaled Scalar plus Vector offset form of gather PRFH instruction might not prefetch from the correct address.

Configurations affected

This erratum affects all configurations.

Conditions

A 32-bit unpacked scaled Scalar plus Vector offset form of gather PRFH instruction is executed from any EL, when *Scalable Vector Extension* (SVE) is enabled and not trapped.

Implications

The affected instruction is a software prefetch which does not affect architectural state in any way (including suppression of any translation faults). Thus, this erratum will not affect the functional operation of the CPU.

Software that relies on explicit prefetches to increase performance might not realize the benefit of the requested prefetches and might instead experience additional cache pollution when executing this instruction targeting incorrect addresses.

Workaround

No workaround is expected to be necessary for this erratum.

3613429
PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability

Status

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

Description

When instructions are not available to be dispatched due to Program Counter Register File (PCRF) fullness, they are counted by the STALL_SLOT_FRONTEND PMU event instead of the STALL_SLOT_BACKEND PMU event.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs whenever instruction fetch is stalled due to PCRF fullness and the PMU is configured to count the STALL_SLOT_FRONTEND or STALL_SLOT_BACKEND events.

Implications

Correlation of STALL_FRONTEND and STALL_SLOT_FRONTEND telemetry might be impacted when the PCRF is often full, because the STALL_FRONTEND PMU event will not count under the same PCRF full conditions.

Workaround

This erratum has no workaround.

3628852
LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled

Status

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

Description

When PE is configured with ERR0CTLR.ED = 0, a load instruction that received data on the CPU AMBA CHI interface with some words marked Poisoned can violate internal visibility requirement.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. PE is configured with ERR0CTLR.ED = 0, disabling Error detection and correction
2. Data requested by a load instruction is received on the CPU AMBA CHI interface with some words marked Poisoned, indicating an uncorrected error has been detected in the system
3. Load consumes non-poisoned words from the returned data.
4. Another PE performs a write to one or more of the bytes consumed by the load

Implications

When the above conditions are met, load instruction might read stale data violating memory ordering requirements.

Workaround

No workaround is expected to be necessary for this erratum.

3631033
Lower priority exception might be reported when abort condition is detected at both stages of translation

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions apply:

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

Implications

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

Workaround

This erratum has no workaround.

3653291
PE might report an unexpected Tag Check fault on a CPY* instruction accessing tagged memory

Status

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

Description

Under certain micro-architectural conditions, a CPY* instruction executing on a *Processing Element* (PE) might incorrectly report a tag check fault.

Configurations Affected

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

Conditions

1. Memory tagging is enabled in precise mode.
2. A CPY* instruction is executed where the destination address is within the last 64 bytes of the TTBR1 address range (0xXXFF_FFFF_FFFF_FFC0 to 0xXXFF_FFF_FFFF_FFFF).
3. The page containing the destination address is writeable.
4. Microarchitectural conditions occur.

Implications

If the previous conditions are met, the PE might incorrectly report a tag check fault.

Workaround

There is no workaround.

3667362
FFR might not capture the lowest faulting memory element

Status

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

Description

Under certain unusual micro-architectural conditions, the *Processing Element* (PE) executing a *Scalable Vector Extension* (SVE) First-fault or Non-fault vector load instruction that fails *Memory Tagging Extension* (MTE) tag check or reads poisoned data might not capture the correct faulting element in the *First Fault Register* (FFR).

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. PE executes an SVE First-fault load instruction with first active element to device memory.
2. PE executes a younger SVE First-fault or Non-fault vector load instruction to normal memory where active element of the Non-fault vector load instruction or non-first active element of the First-fault vector load instruction fails MTE tag check or reads poisoned data.
3. Unusual micro-architectural conditions occur.

Implications

When the above conditions are met, FFR lane corresponding to the lowest faulting memory element might not be set to False.

Workaround

Arm does not expect this issue to occur in realistic code sequences, so no workaround is needed. Please contact Arm for more details.

3676338
PMU events are mis-categorized by not considering the effect of "Taken locally"

Status

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

Description

FEAT_VHE establishes broad use of "Taken locally" as a qualifier that determines which instances of an exception are counted by particular PMU events.

PMU events are mis-categorized by failing to consider "Taken locally", specifically resulting in mis-categorizations between PMU events EXC_UNDEF and EXC_TRAP_OTHER, as well as between PMU events EXC_SVC and EXC_TRAP_OTHER.

Configurations affected

This erratum affects all configurations.

Conditions

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

- When the effective value of HCR_EL2.{E2H,TGE} **is** {1,1}, an exception can increment PMU event 0x008D EXC_TRAP_OTHER, when the exception should instead increment PMU event 0x0081 EXC_UNDEF.
- When the effective value of HCR_EL2.{E2H,TGE} is **NOT** {1,1}, an exception can increment PMU event 0x0081 EXC_UNDEF, when the exception should instead increment PMU event 0x008D EXC_TRAP_OTHER.
- When the effective value of HCR_EL2.{E2H,TGE} is **NOT** {1,1}, executing an SVC instruction can increment PMU event 0x0082 EXC_SVC, when that SVC instruction should instead increment PMU event 0x008D EXC_TRAP_OTHER.

Implications

When the previous conditions are met, PMU event counts might be inaccurate for events 0x0081, 0x0082, and 0x008D.

Workaround

There is no workaround.

3679665
Hang: Test hanging as two LD instructions having the same uid

Status

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

Description

The Instruction-Tag Cache and IData cache enter replay forever (livelock) due to constant parity errors when attempting to access the ITag cache ram.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when there are multiple stuck-at faults for a single ram index (or parity errors every time the Itag cache ram is read, even after ITag cache ram writes to all ways in a ram index). Also, the two ITag pipes must be reading the same index to cause the replay loop.

Implications

If the previous conditions are met, ITag and IData loop forever, delivering no instructions to decode causing livelock.

Workaround

No workaround is expected to be required.

3699915
PE might fail to log a RAS error for L2 data RAM ECC errors

Status

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

Description

Under specific circumstances, the L2 cache might fail to log a corrected or uncorrected ECC error in the PE ERXSTATUS/MISC/ADDR registers.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. Error correction is enabled with ERR0CTLR.ED set to 1.
2. PE is performing simultaneous memory reads to both Device or Normal Non-cacheable and Normal-WriteBack memory.
3. Specific timing conditions occur.
4. PE detects an ECC error in the L2 data RAM.

Implications

If the specified conditions occur, the PE might not report the ECC error detected by the L2.

Note that there is no silent data corruption - any consumers of the data will receive a poison indication along with the data. The issue is a failure to report the error to the RAS error log.

Workaround

No workaround is necessary for this erratum.

3701467
Parity Error in L2BTB can cause failure in breakpoint detection

Status

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

Description

A parity error in L2BTB lookup that has multiple entries in block can cause a breakpoint detection fail for that specific lookup.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following sequence of conditions apply:

1. A breakpoint is set to specific instruction A
2. Instruction A is fetched
3. Instruction A lookup contains parity error in L2BTB

Implications

The breakpoint will be missed, and program execution will errantly continue beyond that breakpoint.

Workaround

There is no workaround.

3705500
VSET.priority might incorrectly designate non-maskable property (NMI)

Status

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

Description

A VSET packet for a virtual interrupt will incorrectly designate the non-maskable property (NMI) if the top 5 bits are all set to 1 (VSET.priority==8'b1111_1xxx), but not all 8 bits are set to 1. The architecture defines when VSET.Grp indicates a Group 1 virtual interrupt. A VSET.priority of 0xFF indicates that the virtual interrupt has the non-maskable property.

Note that it is not possible for any Arm GIC products to be configured to enter the bug state described here.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs if all the following conditions apply:

1. The PE is configured to have 5 bits of priority for virtual interrupts.
2. The PE receives a VSET packet for a virtual interrupt.
3. The priority field of the VSET packet has the top 5 bits set (but not all 8 bits set).

Implications

If the previous conditions are met, the virtual interrupt associated with the VSET packet will be incorrectly designated as an NMI, therefore having the highest possible priority when the intention in this case would be for the virtual interrupt to have the lowest possible priority.

Workaround

A workaround should not be necessary as there are no known configurations which utilize all supported priority bits. However, if a configuration did utilize all supported priority bits, the workaround would be for software to not configure a directly-injected virtual LPI with Priority=0b1111_1xxx.

3705505
ICH_LR<n>_EL2.NMI might incorrectly designate NMI for a Group 0 virtual interrupt

Status

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

Description

When ICH_LR<n>_EL2.NMI is written for a pending or active Group 0 virtual interrupt, the architecture defines the following possible CONSTRAINED UNPREDICTABLE conditions:

- ICH_LR_EL2.NMI is treated as 0 for all purposes other than a direct read of the register.
- The virtual interrupt is presented with superpriority.

If ICH_LR<n>_EL2.NMI is written for a pending or active Group 0 virtual interrupt as previously described, the PE will enter a state which does not satisfy either of the required conditions.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs if all the following conditions apply:

1. The PE is configured to allow virtual interrupts to be sent via the writing of the List Registers (ICH_LR<n>_EL2)
2. ICH_LR<n>_EL2.NMI is set to 1 for a pending or active virtual interrupt
3. The associated ICH_LR<n>_EL2.Group bit is 0

Implications

If the previous conditions are met, the PE will enter a state in which the virtual interrupt will be sent as a Grp0/FIQ without superpriority (NMI=0). However, the internal logic will think that the virtual interrupt is an NMI. This could cause a disruption in the expected interrupt flow as it is not expected that the ICV_NMIAR1_EL1 will be read for a Group 0 virtual interrupt.

Workaround

A workaround should not be necessary as software is not expected to write the ICH_LR<n>_EL2.NMI bit for a Group 0 virtual interrupt.

3706468
ETE counters failing to count events associated with Streaming Mode Compute Unit (SMCU)

Status:

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

Description

In a six PMU Counter configuration, if we program a counter employed by ETE with an SMCU event (part of all implemented PMU events), then there is no event count decrement seen for the window where the event actually occurred.

Configurations affected

This erratum affects six PMU counter configurations only, with ETE enabled.

Conditions

The erratum can occur if all of these conditions apply:

1. Program the CNTEVENT.TYPE and CNTEVENT.SEL in TRCCNTCTLR0 register.
2. Program the initial counter value in TRCCNTVR0 register.
3. RLDSELF in TRCCNTCTLR0 is programmed 0x0.
4. Generate the respective SMCU event.

Implications

If the previous conditions are met, we see that despite the events being counted in PMU CNTR, it remains set to 0 in the TRCCNTVR0 register.

Workaround

There is no workaround.

3708289
Under-counting PMU events involving Streaming Mode Compute Unit (SMCU)

Status

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

Description

In a shared SMCU system, in scenarios where multiple PEs are accessing the same SMCU, some PMU events that count activity within the SMCU might be undercounted. This undercounting can occur in a window after the SMCU has just been switched to a different PE.

Configurations affected

This erratum affects all configurations.

Conditions

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

- SMCU is shared between PEs
- SMCU is switched to a different PE
- PMU event counters have been programmed to count events specific to SMCU (or shared between SMCU and PE)

Implications

If the previous conditions are met, under certain microarchitectural conditions, PMU events associated with SMCU might be undercounted.

Workaround

There is no workaround.

3709839
Incorrect count for PMU event 0x400B (L3D_CACHE_LMISS_RD) might be observed

Status

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

Description

A PRFM instruction might indicate a L3D_CACHE_LMISS_RD PMU event leading to an incorrect count.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. PMU counters are configured to count event 0x400B (L3D_CACHE_LMISS_RD).
2. PRFM instruction causes a refill into the L3D cache.

Implications

If the previous conditions are met, the count indicated by event 0x400B (L3D_CACHE_LMISS_RD) will not match the conditions specified in the Arm Architecture Reference Manual.

Workaround

No workaround is needed unless PMU event 0x400B is required. If a workaround is needed, this erratum can be avoided by counting event 0x8152 (L3D_CACHE_MISS) in place of event 0x400B.

3716859
PMU events SVE_PRED_NOT_FULL_SPEC (evt_0x8079) and SSVE_PRED_NOT_FULL_SPEC (evt_0x3237) count incorrectly

Status

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

Description

When programming PMU event SVE_PRED_NOT_FULL_SPEC (evt_0x8079) or SSVE_PRED_NOT_FULL_SPEC (evt_0x3237), under some scenarios, these events might be overcounted.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

- Enable PMU counters. (PMCR.E, PMCNTENSET, PMSELR).
- PMU counters are configured to count events 0x8079 or 0x3237.
- The PE executes one or more predicated instructions that do not have a vector register as source, nor as destination.

Implications

If the previous conditions are met, under certain microarchitectural conditions, PMU events SVE_PRED_NOT_FULL_SPEC (evt_0x8079) or SSVE_PRED_NOT_FULL_SPEC (evt_0x3237) might be overcounted.

Workaround

No workaround is needed unless PMU event 0x8079 (or 0x3237) is required. If a workaround is needed, this erratum can be avoided by:

- doing a sum of the SVE_PRED_PARTIAL_SPEC (evt_0x8077) and SVE_PRED_EMPTY_SPEC (evt_0x8075) events to get the right count for SVE_PRED_NOT_FULL_SPEC (evt_0x8079).
- doing a sum of the SSVE_PRED_PARTIAL_SPEC (evt_0x3238) and SSVE_PRED_EMPTY_SPEC (evt_0x3235) events to get the right count for SSVE_PRED_NOT_FULL_SPEC (evt_0x3237).

3722310
ESR_ELx.ISV and EDSCR.STATUS does not provide load exclusive syndrome information that could be provided for Software Step or Halting Step

Status

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

Description

For a Software Step exception, ESR_ELx.ISV can not be set to indicate that the ESR_ELx.EX bit is valid under circumstances where ISV could be set.
For a Halting Step debug event, EDSCR.STATUS can be "Halting step, no syndrome" when it could instead be "Halting step, normal".

In each case, no false information is given, but there is missing information that could be helpful to the debugger.

Configurations affected

This erratum affects all configurations.

Conditions

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

- SMCU is shared between PEs.
- SMCU is switched to a different PE.
- An SMSTOP (MSR SVCR (immediate)) instruction is stepped under Software Step or Halting Step.
- That SMSTOP results in a change to either or both of PSTATE.SM and PSTATE.ZA so that the new value of {SM,ZA} is 0b00.

Implications

If the previous conditions are met, under certain microarchitectural conditions, the stepped SMSTOP will result in ESR.ELx.ISV being 0b0 (Software Step), or EDSCR.STATUS being "Halting step, no syndrome" (Halting Step).
The desired behavior would be for ESR_ELx.ISV to be 0b1 (therefore allowing the debugger to interpret ESR_ELx.EX to know that the stepped instruction is not a load exclusive), or for EDSCR.STATUS to be "Halting step, normal" for the same reason.

Workaround

No workaround is necessary.

3727581
PMU events based on a governing predicate are counted incorrectly in streaming mode for operations that do not have a vector register as source or destination

Status:

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

Description:

In streaming mode, predicate operations cause an incorrect count by treating the upper 48 bits of the governing predicate as zero on the following *Performance Monitor Unit* (PMU) events: SVE_PRED_EMPTY_SPEC, SSVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SSVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SSVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC, and SSVE_PRED_NOT_FULL_SPEC.

Configurations affected:

All configurations are affected.

Conditions:

The erratum occurs when all the following conditions apply:

1. PMU counters are configured to count events: SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC, SSVE_PRED_EMPTY_SPEC, SSVE_PRED_FULL_SPEC, SSVE_PRED_PARTIAL_SPEC, or SSVE_PRED_NOT_FULL_SPEC.
2. The core is in streaming mode (PSTATE.SM=1).
3. The *Processing Element* (PE) executes one or more predicated instructions that do not have a vector register as source or destination.

Implications:

When the conditions are met, the erratum would cause an incorrect count on PMU events: SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC, SSVE_PRED_EMPTY_SPEC, SSVE_PRED_FULL_SPEC, SSVE_PRED_PARTIAL_SPEC, or SSVE_PRED_NOT_FULL_SPEC.

Workarounds:

There are no workarounds.

3740433
A Software Step Exception is possible before the entirety of a non aborting CPY* (Memory Copy) instruction completes

Status

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

Configurations affected

This erratum affects all configurations.

Description

A Software Step Exception is possible before the entirety of a non aborting CPY* (Memory Copy) instruction completes.

Conditions

The erratum occurs if the following conditions apply:

1. A Software Step of a CPY* instruction occur
2. Specific micro architectural conditions occur during execution

Implications

Normal software step behavior will be seen while the *Exception Link Register* (ELR) will remain the address of the CPY* instruction. The *Exception Syndrome Register* (ESR) will reflect a single step. Upon continued single stepping, the CPY* instruction will continue from where it stopped.

Workaround

There is no workaround.

3745749
Update to SCR_EL3.EEL2 might inhibit TLB invalidation

Status

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

Description

Under certain conditions, update to SCR_EL3.EEL2 might inhibit certain TLB invalidate instructions.

Configurations affected

All configurations

Conditions

This erratum occurs when all the following conditions apply:

1. PE has lower-EL as Secure. SCR_EL3{NS, EEL2} == {0, 1}.
2. PE allocates a TLB entry for Secure-EL1&0.
3. EL3 transitions lower-EL to Non-Secure and updates EEL2 bit. SCR_EL3{NS, EEL2} == {1, 0}.
4. TLBI ALLE1 is executed targeting Non-Secure translation entries.
5. EL3 transitions lower-EL to Secure. SCR_EL3{NS, EEL2} == {0, 1}.

Implications

After this sequence of conditions, secure EL1 TLB records allocated in condition-2 might not be invalidated by TLB instructions that take a VMID, but will still be hit by address translation. This will cause Secure World to corrupt memory when updating its page tables.

Workaround

No workaround is expected if there is no update to SCR_EL3.EEL2 when changing security states. Otherwise TLBI ALLE1 can be issued whenever security state has changed from Non-Secure to Secure and SCR_EL3.EEL2 has transitioned from 0 to 1 to avoid the erratum.

3750110
CPU fails to inject an IESB after taking a Data Abort resulting from a FEAT_MOPS instruction

Status

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

Description

When the CPU takes an exception, it should inject an implicit error synchronization barrier if SCTLR_ELx.IESB is set. However, data aborts caused by FEAT_MOPS instructions do not cause this barrier to be injected, even when the IESB bit is set. This means outstanding RAS errors will *not* become pending SErrors.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- There are one or more outstanding RAS errors
- The appropriate SCTLR_ELx.IESB bit is set
- A FEAT_MOPS instruction is executed and takes a data abort

Implications

Outstanding RAS errors will be missed upon the data abort caused by the FEAT_MOPS instruction, but will remain outstanding. As long as the SError is not masked, it will still eventually be taken.

Workaround

No workaround is required.

3756951
SPE event 22 might be incorrectly filtered or reported as 0 for load that should have set the event to 1

Status

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

Description

When SPE samples a load, SPE event 22 is collected during the sampled instruction's execution. In certain cases, the sampled load might report 0 for event 22 when the instruction should have actually reported 1 for the event.

Configurations Affected

This erratum affects all configurations.

Conditions:

This erratum might occur when SPE samples a load.

Implications

SPE filtering and reporting of event 22 might be incorrect for some loads.

Workaround

There is no workaround.

3761991
Executing a continuous stream of SME predicated store instrucitons might lead to a denial of service

Status

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

Description

Under certain micro-architectural conditions, executing a continuous stream of SME predicated store instructions might lead to a denial of service.

Configurations affected

This erratum affects all configurations with at least one CME unit in the cluster.

Conditions

The erratum occurs if all the following conditions apply:

1. The core is in SME Streaming mode (PSTATE.SM = 1)
2. The core executes an STLR instruction
3. The core then executes a continuous stream of SME/SVE predicated store instructions with no active elements

Implications

If the previous conditions are met, then the STLR might not be globally observed in finite time.

Workaround

An interrupt on this core will break the sequence of writes and allow the STLR to make forward progress.

3770370
Incorrect fault type might be recorded for Data aborts when VTCR_EL2.HAFT is enabled

Status

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

Description

Under certain conditions, when VTCR_EL2.HAFT is enabled, a stage 2 walk that detects an unsupported atomic fault when trying to update the AF bit in table descriptor might incorrectly report NoTagAccess fault instead of Unsupported atomic hardware update fault in ESR_EL2.DFSC.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

- VTCR_EL2.HAFT is enabled.
- The access is an explicit Allocation Tag read, explicit Allocation Tag Write or Tag Checked access.
- The memory region tagging type for the access is Tagged.
- Stage 2 of translation for output address of stage 1 detects an unsupported atomic hardware update fault when trying to update AF bit in table descriptor.
- bits[5:2] of stage 2 table descriptor are either 4'b0100 or 4'b111x when HCR_EL2.FWB is 1.

Implications

Bits[5:2] of stage 2 table descriptor are not ignored. When the above conditions are met, the PE might report incorrect fault type of NoTagAccess instead of Unsupported atomic hardware update fault in ESR_EL2.DFSC for the data abort exception.

Workaround

When VTCR_EL2.HAFT is enabled, it is expected that stage2 page tables are stored in memory that supports atomics. No workaround is expected to be required.

3770628
PMU event 0x3008 reports incorrect counts

Status

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

Description

The Processing Element (PE) is unable to count event 0x3008, DRAM_ACCESS.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs for all use cases requiring event 0x3008.

Implications

- PMU counters configured to event 0x3008 will report inaccurate event counts.
- Trace functionality is not supported for event 0x3008.

Workaround

There is no workaround for this erratum.

3779319
AMEVCNTR14_EL0 and AMEVTCNTR15_EL0 not wired to events

Status

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

Description

The *Processing Element* (PE) does not have support for AMU auxiliary counters 4 and 5. AMEVTCNTR14_EL0 and AMEVTCNTR15_EL0 will not increment in accordance with their corresponding events, STALL_BACKEND_BUSY_CME (0x3200) and STALL_BACKEND_BUSY_CME_ARB (0x3202), respectively.

Configurations affected

This erratum affects all configurations.

Conditions

PE is attempting to read from AMEVTCNTR14_EL0 or AMEVTCNTR15_EL0.

Implications

All reads of these registers are RAZ.

Workaround

There is no workaround to count these events using the AMU. These events might be counted using PMU event counters.

3782177
Cannot access AMU auxiliary counter enable or disable for auxiliary counters 3, 4, 5

Status

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

Description

The *Processing Element* (PE) does not possess support for external debugger accesses of AMCNTENSET1_EL0 or AMCNTENCLR1_EL0 fields P5, P4, P3.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs for the following conditions:

1. An external debugger is connected to the PE.
2. The external debugger attempts to read from AMCNTENSET1_EL0 or AMCNTENCLR1_EL0.

Implications

External debugger accesses of fields P5, P4, and P3 might incorrectly read zero.

Workaround

The correct value of AMCNTENSET1_EL0 and AMCNTENCLR1_EL0 might be accessed internally via MRS reads of AMCNTENSET1_EL0 or AMCNTENCLR1_EL0.

3830003
The ISV bit value in the Exception Syndrome Register might be incorrect on a software stepped execution of CLRBHB instruction

Status

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

Description

The ISV bit value in the ESR_ELx register is not set to 1 following an exception on software stepped CLRBHB instruction.

Configurations affected

This erratum affects all configurations that implement FEAT_CLRBHB.

Implications

When the above conditions are met, the error syndrome value would be incorrect following an exception on a CLRBHB instruction.

Workaround

There is no workaround.

3836267 SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC PMU events might return incorrect data

Status

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

Description

The following SVE Performance Monitoring Unit (PMU) events should count both non-streaming mode and streaming mode predicate operations.

- SVE_PRED_SPEC,
- SVE_PRED_EMPTY_SPEC,
- SVE_PRED_FULL_SPEC,
- SVE_PRED_PARTIAL_SPEC,
- SVE_PRED_NOT_FULL_SPEC.

However, the listed events only count non-streaming mode operations and do not count streaming mode predicate operations. The RTL incorrectly qualifies the listed PMU events as counting non-streaming mode operations only.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when all the following conditions apply:

- PMU event counters are configured to count events SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, or SVE_PRED_NOT_FULL_SPEC.
- Instructions covered by the preceding events are executed.
- The predicate operation is in streaming mode.

Implications

When all the conditions are met, the SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC PMU events will not be updated and will result in undercounting these events.

Workaround

To achieve the correct result, add the SVE PMU events (SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, SVE_PRED_NOT_FULL_SPEC) to the SSVE PMU events (SSVE_PRED_SPEC, SSVE_PRED_EMPTY_SPEC, SSVE_PRED_FULL_SPEC, SSVE_PRED_PARTIAL_SPEC, SSVE_PRED_NOT_FULL_SPEC).

The listed SSVE PMU events only count streaming mode operations. The sum of the listed SVE PMU events and the listed SSVE PMU events will produce the sum of non-streaming mode and streaming mode operations. This is the correct result for the listed SVE PMU events.

3839259
PMU events triggered by predicate operations count incorrectly

Status

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

Description

Internal *Processing Element* (PE) clock gating might lead to incorrect counts for the following SVE predicated PMU events: SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, and SVE_PRED_NOT_FULL_SPEC.

Configurations affected

This erratum affects all configurations

Conditions

The erratum occurs if all the following conditions apply:

1. PMU counters are configured to count events SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, or SVE_PRED_NOT_FULL_SPEC.
2. Instructions covered by the preceding events are speculatively executed.

Implications

If the previous conditions are met, PMU events SVE_PRED_SPEC, SVE_PRED_EMPTY_SPEC, SVE_PRED_FULL_SPEC, SVE_PRED_PARTIAL_SPEC, and SVE_PRED_NOT_FULL_SPEC might be significantly over counted.

Workaround

This erratum has no workaround.

3877009
Incorrect count for PMU event 0x8285 (L2D_CACHE_PRF) might be observed

Status

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

Description

A hardware or software prefetch operation that accesses the L2D cache might not correctly indicate L2D_CACHE_PRF.

Configurations affected

All configurations

Conditions

The erratum occurs if all the following conditions apply:

1. PMU counters are configured to count event 0x8285 (L2D_CACHE_PRF)
2. A hardware or software prefetch operation accesses the L2D cache

Implications

When the above conditions are met, the count indicated by event 0x8285 (L2D_CACHE_PRF) will not match the conditions specified in the Arm Architecture Reference Manual.

Workaround

There is no workaround.

3895076
PMU event PMU_OVFS is always increased when PMCCNTR_EL0 overflows, independently of PMINTENSET_EL1.C value

Status

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

Description

PMU event PMU_OVFS is always counting overflow from PMCCNTR_EL0 register even while PMINTENSET_EL1.C = 0.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following conditions apply:

- The core executes with PMINTENSET_EL1.C = 0
- PMCCNTR_EL0 overflows

Implications

The following event reports invalid counts:

- 0x400D (PMU_OVFS)

Workaround

There is no workaround for this erratum.

3942089
LDAPUR, LDAPURB, LDAPURH instructions have stricter memory ordering than required

Status

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

Description

LDAPUR instructions execute with full Load-Acquire ordering instead of the relaxed ordering described in the LDAPUR pseudocode. This might cause significant performance degradation in workloads that do not require this stricter memory ordering. Note that this erratum only affects the unscaled versions of LDAPUR (LDAPUR, LDAPURB, LDAPURH), and not LDAPR (LDAPR, LDAPRB, LDAPRH).

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when any LDAPUR, LDAPURB, or LDAPURH instruction is executed.

Implications

The execution of this instruction is architecturally correct, but is less performant than desired. There is no functional problem or architectural violation.

Workaround

No workaround is expected to be necessary for this erratum.

3963302
Incorrect count for PMU event 0x0020 (L2D_CACHE_ALLOCATE) might be observed

Status

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

Description

Certain L2D cache operations might incorrectly indicate L2D_CACHE_ALLOCATE.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following sequence of conditions apply:

1. PMU counters are configured to count event 0x0020 (L2D_CACHE_ALLOCATE).
2. Memory write operation is executed and under certain microarchitectural conditions might overcount event 0x0020 (L2D_CACHE_ALLOCATE).

Implications

When the previous sequence of conditions are met, the count indicated by event 0x0020 (L2D_CACHE_ALLOCATE) will not match the conditions specified in the Arm Architecture Reference Manual.

Workaround

There is no workaround.

3966815
AMU AUX counters 4 and 5 are not counting when PMU is disabled

Status

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

Description

The Activity Monitor Unit (AMU) auxiliary counters 4 and 5 ignore increments for events 0x3200 (STALL_BACKEND_BUSY_CME) and 0x3202 (STALL_BACKEND_BUSY_CME_ARB) when the Performance Monitoring Unit (PMU) is disabled.

Configurations affected

All configurations

Conditions

The erratum occurs if all the following conditions apply:

- AMU auxiliary counters 4 and 5 are enabled and programmed to count
- The Processing Element (PE) is currently using the CME unit
- PMU counting is disabled

Implications

When the previous conditions are met, increments for AMU counter 4 and 5 are not considered.

Workaround

There is no workaround to count these events using the AMU. These events might be counted using PMU event counters.

3980765
Power off transition might deadlock if debug halt request is asserted

Status

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

Description

The *Processing Element* (PE) might deadlock during a transition to the OFF power state if a debug halt request is asserted during the power transition request to off.

Configurations affected

All configurations

Conditions

The erratum might occur under the following sequence of conditions:

1. The PE receives a request to transition from ON to OFF state (via p-channel).
2. A debug halt request is asserted during this OFF transition.

Implications

The PE might become stuck in an intermediate power-down state, blocking OFF state transitions, and leave the system in a potentially unresponsive state.

Workaround

There is no workaround for this erratum.

3984952
STALL_FRONTEND_FLUSH and CPU_CYCLE related PMU events increment during snoop handling in WFx state

Status

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

Description

The following Performance Monitoring Unit (PMU) events increment counters when snoop transactions or other incoming requests that require the cpu clock to be enabled are processed when the cpu clocks are stopped by WFI or WFE instructions:

- 0x8162, STALL_FRONTEND_FLUSH, stall cycles caused by frontend flushes
- 0x3E, STALL_SLOT_FRONTEND, No operation sent for execution on a Slot due to the frontend
- 0x3C, STALL, No operation sent for execution
- 0x23, STALL_FRONTEND, No operation sent for execution due to the frontend.

- 0x11, CPU_CYCLES, total CPU cycle count

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the processor is in a low-power WFx state and receives snoop transactions or other incoming requests that require the cpu clock to be enabled.

Implications

The PMU events 0x8162 (STALL_FRONTEND_FLUSH), 0x3E (STALL_SLOT_FRONTEND), 0x3C (STALL), 0x23 (STALL_FRONTEND) and 0x11 (CPU_CYCLES) increment counters during snoop handling or other incoming requests that require the cpu clock to be enabled even though the core is not actively executing instructions. The architecture defines the behavior as CONSTRAINED UNPREDICTABLE for this condition. Events may reflect higher than expected stall or cycle counts in scenarios with frequent snoop activity during low-power states. This may impact the metrics that use these PMU events (IPC as an example).

Workaround

The PMU event 0x225 (IMP_WFX_CLOCK_CYCLES) counts cycles in wfx and can be subtracted from the impacted pmu counters to exclude cycles in wfx state handling snoops.

3986295
Incorrect exception type reported in ETE exception packet when a SETG* instruction takes a Data Abort exception due to alignment of Xn register

Status

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

Description

When an Embedded Trace Extension (ETE) output trace classifies a Synchronous Data Abort exception due to the alignment of the Xn register while executing a SETG* instruction, the trace incorrectly maps the exception type to an Alignment type. The correct mapping is a Data fault.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following sequence of conditions apply:

1. Not in Debug state
2. A SETG* instruction is executed, with the Xn register not 16-byte aligned
3. ETE Self-hosted tracing is enabled

Implications

When the previous sequence of conditions are met, the PE will record an Exception packet with an exception type of Alignment, rather than Data fault.

Workaround

There is no workaround.

4011447
Incorrect count for PMU event 0x81B4 (DSNP_HIT) might be observed

Status

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

Description

Certain L2 data cache operations, that receive data via a snoop hit in a different cache, might not correctly assert the DSNP_HIT PMU event.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following sequence of conditions apply:

1. PMU counters are configured to count event 0x81B4 (DSNP_HIT).
2. A hardware prefetch or software prefetch operation is executed and receives data via a snoop hit in a different cache.

Implications

When the sequence of conditions is met, the count indicated by event 0x81B4 (DSNP_HIT) will not match the conditions specified in the Arm Architecture Reference Manual.

Workaround

There is no workaround.

4073280
DRPS with invalid SPSR_ELx value incorrectly triggers UNDEF instead of illegal execution state exception

Status

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

Description

When executing a *Debug Restore PE State* (DRPS) instruction with an invalid SPSR_ELx.M value, the *Processing Element* (PE) might trigger an UNDEFINED exception instead of setting PSTATE.IL and generating an illegal execution state exception on the next instruction.

Configurations Affected

All configurations

Conditions

The erratum occurs when all the following conditions apply:

1. The PE is in Debug state and EDSCR.SDD = 1 (root debug disabled).
2. The value of SPSR_ELx.M is invalid or reserved (e.g., 0xD).
3. A DRPS instruction is executed while in Debug state.
4. The PE does not execute any further instructions before exiting Debug state.

Implications

The PE does not raise the expected illegal execution state exception. Instead, the PE might incorrectly raise an UNDEFINED exception.

Workaround

There is no workaround for this erratum.

4103522
Incorrect implementation of CheckSMEEnabled() function in debug state

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Open.

Description

The *Processing Element* (PE) might incorrectly execute SMEAccessTrap() in debug state when SME Access is disabled in CPTR_EL3.

Configurations Affected

This erratum affects all configurations that support FEAT_SME.

Conditions

This erratum occurs when all the following conditions apply:

1. The PE is in Debug State (Halted).
2. CPTR_EL3.ESM is programmed to 0.
3. EDSCR.SDD is programmed to 1.
4. Current *Exception Level* (EL) is not EL3.
5. Current Security State is not SS_Secure.

Implications

When all the above conditions are met, a FEAT_SME instruction is incorrectly executed because it triggers SMEAccessTrap() to the current EL, instead of taking an UNDEFINED exception.

Workaround

This erratum has no workaround.

4106082
ETE and ELA timestamps are not accurate for core

Status

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

Description

Embedded Trace Extension (ETE) and Embedded Logic Analyzer (ELA) timestamps are not transported correctly for a core that is in the OFF_EMU power mode.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core is in the OFF or OFF_EMU power mode.
2. One of the following conditions is true:
    - Timestamp trace insertion is enabled in ETE when TRCCONFIGR.TS = 1.
    - Timestamp trace insertion is enabled in ELA by setting TIMECTRL.TSEN.

Implications

During debug, any timestamp inserted into the trace stream will be inaccurate while in the OFF_EMU power mode.

Workaround

Debug software that is decoding trace output should be aware that the timestamps might be incorrect during periods in the OFF_EMU power mode, and ignore such timestamps.

4128618
Incorrect sampling of SPE partial and empty predicate values

Status

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

Description

Under certain circumstances, the Statistical Profiling Extension (SPE) events E[17] "Partial predicate" and E[18] "Empty predicate" might not be captured as defined in the Arm® Architecture Reference Manual for A-profile architecture.

Configurations Affected

All configurations

Conditions:

SPE samples a Scalable Vector Extension (SVE) instruction with predicate operands.

Implications

When the above conditions are met, the SPE events E[17] "Partial predicate" and E[18] "Empty predicate" will be captured as 0 for the given sample.

Workaround

There is no workaround.

Proprietary notice

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 for a product in development and is not final.

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.