Arm Cortex-A715 (MP148)

Software Developer Errata Notice

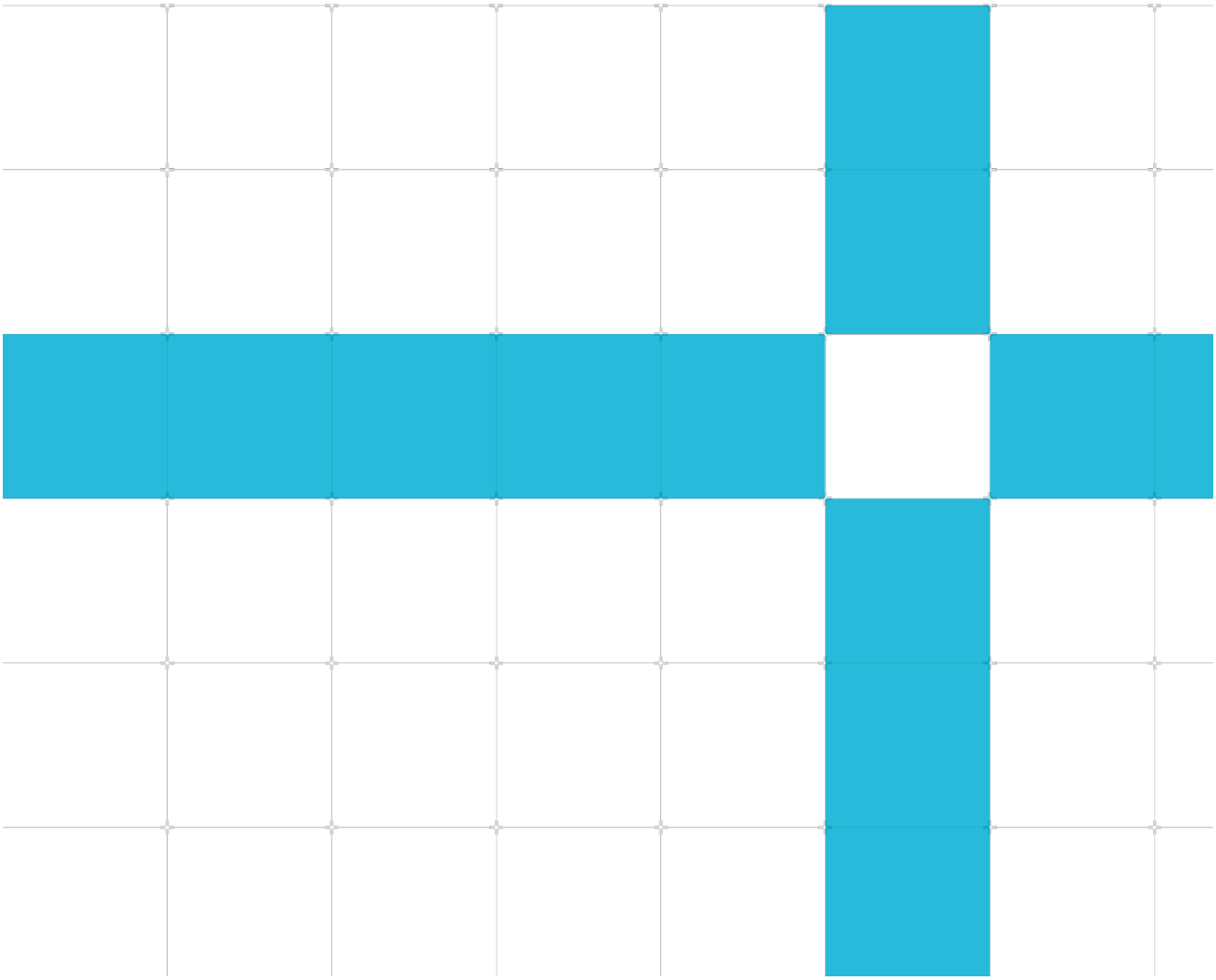
Date of issue: 30-Sep-2022

Non-Confidential

Document version: v10.0

Document ID: SDEN-2148827

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



Non-confidential proprietary notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality status

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

Unrestricted Access is an Arm internal classification.

Product status

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Cortex-A715 (MP148), create a ticket on **https://support.developer.arm.com**.

To provide feedback on the document, fill the following survey: **https://developer.arm.com/documentation-feedback-survey**.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

If you find offensive language in this document, please email **terms@arm.com**.

Contents

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 value or a set bit indicates that the erratum is fixed in this part.

REVIDR_EL1[23:0] = 0x1	2357487	A stream of Stashes, Cache Maintenance Operations, MTE-checked stores or atomics may cause the CPU to deadlock or lose coherency

REVIDR_EL1[23:0] = 0x2	2357487	A stream of Stashes, Cache Maintenance Operations, MTE-checked stores or atomics may cause the CPU to deadlock or lose coherency

REVIDR_EL1[23:0] = 0x3	2357487	A stream of Stashes, Cache Maintenance Operations, MTE-checked stores or atomics may cause the CPU to deadlock or lose coherency
	2418468	Crossing 4K loads might cause ordering violation or silent corruption in MTE asymmetric mode

REVID[2] is set	2409570	Store-Release instructions might not be observed in the correct order

Note that there is no change to the MIDR_EL1 which remains at r1p0 but the REVIDR is updated to indicate which errata are corrected. 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.

30-Sep-2022: Changes in document version v10.0

ID	Status	Area	Category	Summary
2645198	Updated	Programmer	Category B	Values of FAR_ELx and ESR_ELx on an Instruction Abort might not be consistent with the instruction that generated the fault
2238597	Updated	Programmer	Category C	A MTE checked store might report a speculative SError in case of poisoned tags
2293949	Updated	Programmer	Category C	Instruction sampling bias exists in SPE implementation
2301922	Updated	Programmer	Category C	Some PMU events might have incorrect values
2305209	Updated	Programmer	Category C	Executing an ESB might fail to clear a masked Virtual SError Interrupt
2318005	Updated	Programmer	Category C	SPE Issue Latency counter might be incorrect on sampled operations generating an exception
2372608	Updated	Programmer	Category C	Double bit error in L2 Tag RAM might lead to deadlock or protocol violation
2422640	Updated	Programmer	Category C	A continuous flow of snoops and atomics might prevent a store from becoming visible in finite time
2455253	Updated	Programmer	Category C	Crossing 4KB load might report incorrect FFR index
2460643	Updated	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails
2625935	Updated	Programmer	Category C	UNDEFINED exceptions due to EDSCR.SDD might be prioritized over EL2 traps caused by HCR_EL2.TIDCP
2631695	Updated	Programmer	Category C	Incorrect transition to Software-Step Active state after OSLAR_EL1.OSLK write followed by ERET
2701325	New	Programmer	Category C	The PE might not respond to APB accesses to some registers in OFF_EMU
2709709	New	Programmer	Category C	Interrupts might be taken following a Context Synchronization Event when MDSR_EL1.INTdis or EDSCR.INTdis are set
2712353	New	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect
2731260	New	Programmer	Category C	Some PMU events might have incorrect values

24-Jun-2022: Changes in document version v9.0

ID	Status	Area	Category	Summary
2645198	New	Programmer	Category B	Values of FAR_ELx and ESR_ELx on an Instruction Abort might not be consistent with the instruction that generated the fault
2389413	Updated	Programmer	Category C	Some PMU events might have incorrect values
2422640	Updated	Programmer	Category C	A continuous flow of snoops and atomics might prevent a store from becoming visible in finite time
2661229	New	Programmer	Category C	ESR_ELx for a Breakpoint exception might be incorrect if breakpoints are concurrently disabled through APB

25-May-2022: Changes in document version v8.0

ID	Status	Area	Category	Summary
2357487	Updated	Programmer	Category A	A stream of Stashes, Cache Maintenance Operations, MTE-checked stores or atomics may cause the CPU to deadlock or lose coherency
2330952	Updated	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing
2331818	Updated	Programmer	Category B	Store might never complete a coherency operation, leading to deadlock
2344187	Updated	Programmer	Category B	Slow MRS instruction can get corrupted data
2376701	Updated	Programmer	Category B	LDFF/LDNF can generate a corrupted First Fault Register
2409570	Updated	Programmer	Category B	Store-Release instructions might not be observed in the correct order
2413290	New	Programmer	Category B	Enabling the Statistical Profiling Extension might lead to deadlock
2418468	Updated	Programmer	Category B	Crossing 4K loads might cause ordering violation or silent corruption in MTE asymmetric mode
2420947	Updated	Programmer	Category B	Executing a store and store release on the same cacheline might lead to corruption
2429384	Updated	Programmer	Category B	Executing uncacheable loads might lead to deadlock
2561034	Updated	Programmer	Category B	Speculatively executing 4KB-crossing loads might lead to data corruption
2615073	New	Programmer	Category B	CPU might violate restrictions on instructions fetches when all stages of translation are disabled
2293949	Updated	Programmer	Category C	Instruction sampling bias exists in SPE implementation
2305209	Updated	Programmer	Category C	Executing an ESB might fail to clear a masked Virtual SError Interrupt
2310079	Updated	Programmer	Category C	PMBSR_EL1 might report a Data Abort instead of a Buffer Management Event when PMBPTR_EL1 reaches PMBLIMITR_EL1
2312234	Updated	Programmer	Category C	Set/Way CMO instructions might not affect a cache line stored in the L1 victim cache
2318005	Updated	Programmer	Category C	SPE Issue Latency counter might be incorrect on sampled operations generating an exception
2328270	Updated	Programmer	Category C	Trace Buffer Extension might fail to complete all pending accesses following a Data Synchronization Barrier or Distributed Virtual Memory Sync

ID	Status	Area	Category	Summary
2354488	Updated	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect
2372608	Updated	Programmer	Category C	Double bit error in L2 Tag RAM might lead to deadlock or protocol violation
2385318	Updated	Programmer	Category C	PMCFGR.EX and PMCR_EL0.X should be 0
2387304	Updated	Programmer	Category C	Checked crossing 64B boundary load might fail to report a tag check fault
2389413	Updated	Programmer	Category C	Some PMU events might have incorrect values
2411188	Updated	Programmer	Category C	ESR_ELx.ISV might be incorrect when software-stepping an MSR DAIF instruction
2422640	Updated	Programmer	Category C	A continuous flow of snoops and atomics might prevent a store from becoming visible in finite time
2424427	Updated	Programmer	Category C	Uncacheable 32-bytes loads might fail to report poison information
2426380	Updated	Programmer	Category C	Cacheable load might return corrupted data upon single-bit error in L1 data cache tag RAM
2428886	Updated	Programmer	Category C	ERXSTATUS_EL1.OF might incorrectly be set in case of multiple errors
2455253	Updated	Programmer	Category C	Crossing 4KB load might report incorrect FFR index
2460643	Updated	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails
2574284	New	Programmer	Category C	Software-step with no instruction step after debug-exit from EL0 with a bad mode in DSPSR
2625935	New	Programmer	Category C	UNDEFINED exceptions due to EDSCR.SDD might be prioritized over EL2 traps caused by HCR_EL2.TIDCP
2631722	New	Programmer	Category C	RAMINDEX operations accessing the L1 Data cache might fail to return the correct result
2631695	New	Programmer	Category C	Incorrect transition to Software-Step Active state after OSLAR_EL1.OSLK write followed by ERET

26-Apr-2022: Changes in document version v7.0

ID	Status	Area	Category	Summary
2429384	New	Programmer	Category B	Executing uncacheable loads might lead to deadlock
2561034	New	Programmer	Category B	Speculatively executing 4KB-crossing loads might lead to data corruption
2293949	New	Programmer	Category C	Instruction sampling bias exists in SPE implementation
2354488	Updated	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect
2385318	New	Programmer	Category C	PMCFGR.EX and PMCR_EL0.X should be 0
2389413	Updated	Programmer	Category C	Some PMU events might have incorrect values
2411188	New	Programmer	Category C	ESR_ELx.ISV might be incorrect when software-stepping an MSR DAIF instruction
2422640	New	Programmer	Category C	A continuous flow of snoops and atomics might prevent a store from becoming visible in finite time
2455253	New	Programmer	Category C	Crossing 4KB load might report incorrect FFR index
2460643	New	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails

27-Jan-2022: Changes in document version v6.0

ID	Status	Area	Category	Summary
2330952	New	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing
2344187	New	Programmer	Category B	Slow MRS instruction can get corrupted data
2376701	New	Programmer	Category B	LDFF/LDNF can generate a corrupted First Fault Register
2409570	New	Programmer	Category B	Store-Release instructions might not be observed in the correct order
2418468	New	Programmer	Category B	Crossing 4K loads might cause ordering violation or silent corruption in MTE asymmetric mode
2420947	New	Programmer	Category B	Executing a store and store release on the same cacheline might lead to corruption
2354488	Updated	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect
2387304	New	Programmer	Category C	Checked crossing 64B boundary load might fail to report a tag check fault
2389413	New	Programmer	Category C	Some PMU events might have incorrect values
2424427	New	Programmer	Category C	Uncacheable 32-bytes loads might fail to report poison information
2426380	New	Programmer	Category C	Cacheable load might return corrupted data upon single-bit error in L1 data cache tag RAM
2428886	New	Programmer	Category C	ERXSTATUS_EL1.OF might incorrectly be set in case of multiple errors

30-Nov-2021: Changes in document version v5.0

ID	Status	Area	Category	Summary
2357487	New	Programmer	Category A	A stream of Stashes, Cache Maintenance Operations, MTE-checked stores or atomics may cause the CPU to deadlock or lose coherency
2331818	New	Programmer	Category B	Store might never complete a coherency operation, leading to deadlock
2328270	New	Programmer	Category C	Trace Buffer Extension might fail to complete all pending accesses following a Data Synchronization Barrier or Distributed Virtual Memory Sync
2354488	New	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect
2372608	New	Programmer	Category C	Double bit error in L2 Tag RAM might lead to deadlock or protocol violation

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

ID	Status	Area	Category	Summary
2248167	Updated	Programmer	Category B	ESR_ELx and FAR_ELx might report incorrect information on Instruction Abort when hardware breakpoints are enabled
2275754	New	Programmer	Category B	Speculatively executed store behind constantly mis-predicted branch might prevent forward progress of synchronization primitives in other PEs
2284544	New	Programmer	Category B	Exclusive store might deadlock under very specific timing conditions
2285536	New	Programmer	Category B	TRBE might generate corrupt trace data and/or fail to complete all pending accesses following a DSB or DVM Sync
2285473	New	Programmer	Category B	A LDR executed between two Non-Cacheable LDRs to the same 64B boundary might fail to apply ordered-before semantics to said instructions
2287512	New	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag, Idata, and TLB RAMs
2292761	New	Programmer	Category B	Executing code in a context with multiple VAs mapped to the same PA might lead to an opcode corruption
2302347	New	Programmer	Category B	Write-After-Write ordering might be violated in presence of STLR
2317812	New	Programmer	Category B	Sequential consistency might not be respected with a non cacheable Store-Release
2180579	Updated	Programmer	Category C	Load/Store instructions might fail to report double-bit ECC errors, and/or External aborts, and/or cause loss of poison information Status
2227533	New	Programmer	Category C	APB accesses to unallocated addresses might generate an PSLVERR response
2247884	New	Programmer	Category C	A continuous stream of stores might prevent an older store from becoming visible in finite time
2262629	New	Programmer	Category C	Checked store in precise mode might deadlock in case of external abort
2268885	New	Programmer	Category C	PE might generate spurious Branch Target exceptions when dynamically changing the GP bit in page tables
2276160	New	Programmer	Category C	PMU event DTLB_WALK might incorrectly count some instructions in case of EPD/E0PD translation fault
2301922	New	Programmer	Category C	Some PMU events might have incorrect values
2305209	New	Programmer	Category C	Executing an ESB might fail to clear a masked Virtual SError Interrupt
2310079	New	Programmer	Category C	PMBSR_EL1 might report a Data Abort instead of a Buffer Management Event when PMBPTR_EL1 reaches PMBLIMITR_EL1
2312234	New	Programmer	Category C	Set/Way CMO instructions might not affect a cache line stored in the L1 victim cache
2318005	New	Programmer	Category C	SPE Issue Latency counter might be incorrect on sampled operations generating an exception

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

ID	Status	Area	Category	Summary
2180026	New	Programmer	Category B	Reset value of MDCR_EL2.HPMN does not reflect the number of implemented PMU counters
2238661	New	Programmer	Category B	In MTE precise mode, Load with writeback can cause the PE to deadlock
2239006	New	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation.
2248167	New	Programmer	Category B	ESR_ELx and FAR_ELx might report incorrect information on Instruction Abort when hardware breakpoints are enabled
2268070	New	Programmer	Category B	Data Cache Clean operations by VA to the Point of Coherency, Point of Persistence, or Point of Deep Persistence might cause the PE to deadlock
2182762	New	Programmer	Category C	Direct RAM Access read on IData RAM might return corrupted data
2220769	New	Programmer	Category C	Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work
2238597	New	Programmer	Category C	A MTE checked store might report a speculative SError in case of poisoned tags
2246816	New	Programmer	Category C	Disabling MPMM through Utility Bus can cause a deadlock
2252693	New	Programmer	Category C	MPAM information sent for TBE addresses might not use TBE owning translation regime
2259398	New	Programmer	Category C	Software-step not done after debug-exit with a bad mode in DSPSR

09-Jul-2021: Changes in document version v2.0

ID	Status	Area	Category	Summary
2160316	New	Programmer	Category C	Trace data might miss some packets or contain incorrect packets when enabled through the APB interface
2168152	New	Programmer	Category C	Persistent errors on SVE First-fault instructions might result in deadlock under specific conditions
2171772	New	Programmer	Category C	A double-bit ECC error or poison on tags on a First-fault load might result in a Synchronous Tag Check Fault instead of a Synchronous External abort
2180579	New	Programmer	Category C	Load/Store instructions might fail to report double-bit ECC errors, and/or External aborts, and/or cause loss of poison information Status
2187565	New	Programmer	Category C	A double-bit ECC error on the L2 TQ RAM for a store might fail to mark the data as poisoned
2189618	New	Programmer	Category C	ELR_ELx might contain an incorrect value on exceptions taken from PC 0xFFFF_0000_0000_0000
2238671	New	Programmer	Category C	SVE PRFM instructions fail to prefetch at the expected address
2238988	New	Programmer	Category C	A continuous stream of stores might prevent forward progress of a coherency operation

30-Apr-2021: 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
2357487	Programmer	Category A	A stream of Stashes, Cache Maintenance Operations, MTE-checked stores or atomics may cause the CPU to deadlock or lose coherency	r0p0, r1p0	r1p1
2645198	Programmer	Category B	Values of FAR_ELx and ESR_ELx on an Instruction Abort might not be consistent with the instruction that generated the fault	r0p0, r1p0, r1p1	r1p2
2615073	Programmer	Category B	CPU might violate restrictions on instructions fetches when all stages of translation are disabled	r0p0, r1p0	r1p1
2561034	Programmer	Category B	Speculatively executing 4KB-crossing loads might lead to data corruption	r1p0	r1p1
2429384	Programmer	Category B	Executing uncacheable loads might lead to deadlock	r0p0, r1p0	r1p1
2420947	Programmer	Category B	Executing a store and store release on the same cacheline might lead to corruption	r1p0	r1p1
2418468	Programmer	Category B	Crossing 4K loads might cause ordering violation or silent corruption in MTE asymmetric mode	r0p0, r1p0	r1p1
2413290	Programmer	Category B	Enabling the Statistical Profiling Extension might lead to deadlock	r1p0	r1p1
2409570	Programmer	Category B	Store-Release instructions might not be observed in the correct order	r0p0, r1p0	r1p1
2376701	Programmer	Category B	LDFF/LDNF can generate a corrupted First Fault Register	r0p0, r1p0	r1p1
2344187	Programmer	Category B	Slow MRS instruction can get corrupted data	r0p0, r1p0	r1p1
2331818	Programmer	Category B	Store might never complete a coherency operation, leading to deadlock	r0p0, r1p0	r1p1
2330952	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing	r0p0, r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2317812	Programmer	Category B	Sequential consistency might not be respected with a non cacheable Store-Release	r0p0	r1p0
2302347	Programmer	Category B	Write-After-Write ordering might be violated in presence of STLR	r0p0	r1p0
2292761	Programmer	Category B	Executing code in a context with multiple VAs mapped to the same PA might lead to an opcode corruption	r0p0	r1p0
2287512	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag, Idata, and TLB RAMs	r0p0	r1p0
2285473	Programmer	Category B	A LDR executed between two Non-Cacheable LDRs to the same 64B boundary might fail to apply ordered-before semantics to said instructions	r0p0	r1p0
2285536	Programmer	Category B	TRBE might generate corrupt trace data and/or fail to complete all pending accesses following a DSB or DVM Sync	r0p0	r1p0
2284544	Programmer	Category B	Exclusive store might deadlock under very specific timing conditions	r0p0	r1p0
2275754	Programmer	Category B	Speculatively executed store behind constantly mis-predicted branch might prevent forward progress of synchronization primitives in other PEs	r0p0	r1p0
2268070	Programmer	Category B	Data Cache Clean operations by VA to the Point of Coherency, Point of Persistence, or Point of Deep Persistence might cause the PE to deadlock	r0p0	r1p0
2248167	Programmer	Category B	ESR_ELx and FAR_ELx might report incorrect information on Instruction Abort when hardware breakpoints are enabled	r0p0	r1p0
2239006	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation.	r0p0	r1p0
2238661	Programmer	Category B	In MTE precise mode, Load with writeback can cause the PE to deadlock	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
2180026	Programmer	Category B	Reset value of MDCR_EL2.HPMN does not reflect the number of implemented PMU counters	r0p0	r1p0
2731260	Programmer	Category C	Some PMU events might have incorrect values	r0p0, r1p0, r1p1, r1p2	Open
2712353	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect	r1p0, r1p1, r1p2	Open
2709709	Programmer	Category C	Interrupts might be taken following a Context Synchronization Event when MDSR_EL1.INTdis or EDSCR.INTdis are set	r0p0, r1p0, r1p1, r1p2	Open
2701325	Programmer	Category C	The PE might not respond to APB accesses to some registers in OFF_EMU	r0p0, r1p0, r1p1	r1p2
2661229	Programmer	Category C	ESR_ELx for a Breakpoint exception might be incorrect if breakpoints are concurrently disabled through APB	r0p0, r1p0	r1p1
2631695	Programmer	Category C	Incorrect transition to Software-Step Active state after OSLAR_EL1.OSLK write followed by ERET	r0p0, r1p0, r1p1, r1p2	Open
2631722	Programmer	Category C	RAMINDEX operations accessing the L1 Data cache might fail to return the correct result	r0p0, r1p0	r1p1
2625935	Programmer	Category C	UNDEFINED exceptions due to EDSCR.SDD might be prioritized over EL2 traps caused by HCR_EL2.TIDCP	r0p0, r1p0, r1p1, r1p2	Open
2574284	Programmer	Category C	Software-step with no instruction step after debug-exit from EL0 with a bad mode in DSPSR	r0p0, r1p0	r1p1
2460643	Programmer	Category C	Checked stores in precise mode might fail to report tag check fails	r0p0, r1p0, r1p1, r1p2	Open
2455253	Programmer	Category C	Crossing 4KB load might report incorrect FFR index	r1p0, r1p1, r1p2	Open
2428886	Programmer	Category C	ERXSTATUS_EL1.OF might incorrectly be set in case of multiple errors	r0p0, r1p0	r1p1
2426380	Programmer	Category C	Cacheable load might return corrupted data upon single-bit error in L1 data cache tag RAM	r0p0, r1p0	r1p1
2424427	Programmer	Category C	Uncacheable 32-bytes loads might fail to report poison information	r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2422640	Programmer	Category C	A continuous flow of snoops and atomics might prevent a store from becoming visible in finite time	r0p0, r1p0, r1p1, r1p2	Open
2411188	Programmer	Category C	ESR_ELx.ISV might be incorrect when software-stepping an MSR DAIF instruction	r0p0, r1p0	r1p1
2389413	Programmer	Category C	Some PMU events might have incorrect values	r0p0, r1p0	r1p1
2387304	Programmer	Category C	Checked crossing 64B boundary load might fail to report a tag check fault	r0p0, r1p0	r1p1
2385318	Programmer	Category C	PMCFGR.EX and PMCR_EL0.X should be 0	r0p0, r1p0	r1p1
2372608	Programmer	Category C	Double bit error in L2 Tag RAM might lead to deadlock or protocol violation	r0p0, r1p0, r1p1, r1p2	Open
2354488	Programmer	Category C	Information in some Statistical Profiling Extension packets might be incorrect	r1p0	r1p1
2328270	Programmer	Category C	Trace Buffer Extension might fail to complete all pending accesses following a Data Synchronization Barrier or Distributed Virtual Memory Sync	r1p0	r1p1
2318005	Programmer	Category C	SPE Issue Latency counter might be incorrect on sampled operations generating an exception	r1p0, r1p1, r1p2	Open
2312234	Programmer	Category C	Set/Way CMO instructions might not affect a cache line stored in the L1 victim cache	r0p0, r1p0	r1p1
2310079	Programmer	Category C	PMBSR_EL1 might report a Data Abort instead of a Buffer Management Event when PMBPTR_EL1 reaches PMBLIMITR_EL1	r1p0	r1p1
2305209	Programmer	Category C	Executing an ESB might fail to clear a masked Virtual SError Interrupt	r0p0, r1p0, r1p1, r1p2	Open
2301922	Programmer	Category C	Some PMU events might have incorrect values	r0p0	r1p0
2293949	Programmer	Category C	Instruction sampling bias exists in SPE implementation	r1p0, r1p1, r1p2	Open
2276160	Programmer	Category C	PMU event DTLB_WALK might incorrectly count some instructions in case of EPD/E0PD translation fault	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
2268885	Programmer	Category C	PE might generate spurious Branch Target exceptions when dynamically changing the GP bit in page tables	r0p0	r1p0
2262629	Programmer	Category C	Checked store in precise mode might deadlock in case of external abort	r0p0	r1p0
2259398	Programmer	Category C	Software-step not done after debug-exit with a bad mode in DSPSR	r0p0	r1p0
2252693	Programmer	Category C	MPAM information sent for TBE addresses might not use TBE owning translation regime	r0p0	r1p0
2247884	Programmer	Category C	A continuous stream of stores might prevent an older store from becoming visible in finite time	r0p0	r1p0
2246816	Programmer	Category C	Disabling MPMM through Utility Bus can cause a deadlock	r0p0	r1p0
2238988	Programmer	Category C	A continuous stream of stores might prevent forward progress of a coherency operation	r0p0	r1p0
2238597	Programmer	Category C	A MTE checked store might report a speculative SError in case of poisoned tags	r0p0, r1p0, r1p1, r1p2	Open
2238671	Programmer	Category C	SVE PRFM instructions fail to prefetch at the expected address	r0p0	r1p0
2227533	Programmer	Category C	APB accesses to unallocated addresses might generate an PSLVERR response	r0p0	r1p0
2220769	Programmer	Category C	Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work	r0p0	r1p0
2189618	Programmer	Category C	ELR_ELx might contain an incorrect value on exceptions taken from PC 0xFFFF_0000_0000_0000	r0p0	r1p0
2187565	Programmer	Category C	A double-bit ECC error on the L2 TQ RAM for a store might fail to mark the data as poisoned	r0p0	r1p0
2182762	Programmer	Category C	Direct RAM Access read on IData RAM might return corrupted data	r0p0	r1p0
2180579	Programmer	Category C	Load/Store instructions might fail to report double-bit ECC errors, and/or External aborts, and/or cause loss of poison information Status	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
2171772	Programmer	Category C	A double-bit ECC error or poison on tags on a First-fault load might result in a Synchronous Tag Check Fault instead of a Synchronous External abort	r0p0	r1p0
2168152	Programmer	Category C	Persistent errors on SVE First-fault instructions might result in deadlock under specific conditions	r0p0	r1p0
2160316	Programmer	Category C	Trace data might miss some packets or contain incorrect packets when enabled through the APB interface	r0p0	r1p0

Errata descriptions

Category A

2357487
A stream of stashes, cache maintenance operations, MTE-checked stores or atomics might cause the core to deadlock or lose coherency

Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0, and r1p0. Fixed in r1p1.

Description

A stream of stashing operations, cache maintenance to the Point of Persistence, or Point of Deep Persistence, MTE-checked store operations in imprecise/asymmetric mode (SCTLR_ELx.TCF=0b10 or 0b11), or MTE-checked atomic operations in imprecise mode (SCTLR_ELx.TCF=0b10) can cause the core to deadlock or lose coherency.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs when the core reaches a number of outstanding transactions on the CHI interface larger than 256 for any of the following groups of operations:

- Stash operations.
- Cache maintenance operations to the Point of Persistence or Point of Deep Persistence, MTE-checked full cache line writes, MTE-checked atomic transactions.

Implications

If the conditions are met, under specific timing conditions, the PE can deadlock or lose coherency.

Workaround

For r1p0, this erratum has no workaround.

For r0p0, this erratum can be avoided by applying the three following actions. The first action is required in all cases, and the second and third actions depend on the DSU configuration.

1/ Use the following sequence to configure the core to wait the completion of all previous memory operations before executing any cache maintenance operations to the Point of Persistence, or Point of Deep Persistence:

```
MOV x0, #4
MSR s3_6_c15_c8_0, x0
ISB
LDR x0, =0xd50b_7c00
MSR s3_6_c15_c8_2, x0
LDR x0, =0xffff_fe00
MSR s3_6_c15_c8_3, x0
MOV x1, #0
ORR x1, x1, #1<<0
ORR x1, x1, #3<<4
ORR x1, x1, #0xf<<6
ORR x1, x1, #1<<20
ORR x1, x1, #1<<33
MSR s3_6_c15_c8_1, x1
ISB
```

2/ If the DSU has one of the following configurations:

- NUM_L3_SLICES=4, NUM_LTDBS=64
- NUM_L3_SLICES=8, NUM_LTDBS=32
- NUM_L3_SLICES=8, NUM_LTDBS=48
- NUM_L3_SLICES=8, NUM_LTDBS=64

then perform the two following actions:

- set bit[1] of CPUECTLR_EL1, to disable stashing operations, and
- set bit[44] of CPUACTLR_EL1, to disable PRFM operations.

3/ If the DSU is configured with BROADCASTMTE=1, set bit[17] of CPUACTLR2_EL1 to force all atomic operations to be executed in the L1 cache and either:

- set bit[23] of CPUACTLR2_EL1, to force MTE-checked stores to write in the L1 cache, or
- write bits[19:13] of CPUECTLR_EL1 to 0b0111111, to prevent all stores from writing beyond the L2 cache.

Category A (rare)

There are no errata in this category.

Category B

2645198
Values of FAR_ELx and ESR_ELx on an Instruction Abort might not be consistent with the instruction that generated the fault

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

Description

If an instruction (F1) is architecturally executed on a PE (PE0) while its mapping is being changed by another PE (PE1), an Instruction Abort generated on PE0 for an instruction F2 that is in program order after F1, might report ESR_ELx and FAR_ELx values consistent with a fault being taken on F1.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum happens when all of the following conditions are met:

- PE0 architecturally executes an instruction F1
- PE1 concurrently modifies the mapping for F1 in a way that fetching from this mapping would cause a Permission Fault
- PE0 takes any of the following exceptions for an instruction F2 in program order after F1:
  - an Instruction Abort because of a Permission Fault
  - a Breakpoint exception, and F1 and F2 are within the same 32B-aligned, 32B granule
- There is no Context Synchronization Event between the execution of F1 and F2.

Implications

If the previous conditions are met:

- The Instruction Abort or Breakpoint exception generated for F2 is routed consistently with the cause of the exception.
- The values of ESR_ELx and FAR_ELx for this exception are consistent with an Instruction Abort caused by a Permission Fault generated by fetching from the mapping for F1 installed by P1.
- ELR_ELx points to F2

Workaround

This erratum can be avoided by ensuring that break-before-make is used when transitioning a page from Executable to Non-Executable.

2615073
CPU might violate restrictions on instructions fetches when all stages of translation are disabled

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, and r1p0. Fixed in r1p1.

Description

CPU might violate restrictions on instructions fetches when all stages of translation are disabled.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- All stages of translation are disabled for the current execution context
- For any 32-bit-aligned, 32-bit memory block between the current architectural PC and the end of the next 4KB granule:
  - the memory block contains an AArch64 branch opcode B1
  - the memory block is not accessed as part of a simple sequential execution of the program
  - This might happen, for example, if there is data placed after instructions in the same 4KB granule of memory.
- For r1p0 only, either:
  - PSTATE.SSBS=1
  - PSTATE.SSBS=0, and the 32-byte-aligned, 32-byte granule that contains B1 also contains any of the following architecturally-executed instructions:
    - WRFFR
    - Any Load instruction that causes either a Data Abort, an SP Alignment Fault, or a Watchpoint exception

Implications

When the above conditions are met, the PE might access memory locations as a result of instruction fetches where none of the conditions described in the Arm Architecture Reference Manual for A-profile architecture (issue H.a, section: D5.2.9 The effects of disabling a stage of address translation) are met:

- The memory location is in the same block of memory as, or in the next contiguous block of memory to, an instruction that a simple sequential execution of the program either requires to be fetched

now or has required to be fetched since the last reset.
- The memory location is the target of a direct branch that a simple sequential execution of the program would have taken since the most recent of:
  - the last reset
  - the last synchronization of instruction cache maintenance targeting the address of the branch instruction

Workaround

- For r0p0 there is no workaround. If a software workaround is required, contact Arm for more details.
- For r1p0, this erratum can be avoided by doing all of the following:
  - setting PSTATE.SSBS=0 when all stages of translation are disabled
  - ensuring that any 32-byte-aligned, 32-byte granule that contains B1 does not contain any of the following architecturally-executed instructions:
    - WRFFR
    - Any Load instruction that causes either a Data Abort, an SP Alignment Fault, or a Watchpoint exception

2561034
Speculatively executing 4KB-crossing loads might lead to data corruption

Status

Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

Description

Executing loads crossing a 4KB boundary might cause such loads to return incorrect data, leading to data corruption.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- The processing element (PE) speculatively executes a 4KB-crossing load R1.
- Other rare microarchitectural conditions occur.

Implications

If the previous conditions are met and under very specific timing conditions, it is possible for R1 to return corrupted data.

Workaround

This erratum can be avoided by setting bit[26] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<26
MSR S3_0_C15_C1_1, x0
```

Setting this bit is not expected to have a significant performance impact.

2429384
Executing uncacheable loads might lead to deadlock

Status

Fault Type: Programmer Category B.
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

Executing multiple uncacheable loads in a row might lead to deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- [On r0p0] The *Processing Element* (PE) executes an uncacheable SVE gather load, R1.
- [On r1p0] The PE executes any uncacheable load, R1.
- The PE executes a second uncacheable load, R2.

Implications

If the above conditions are met and under very specific timing conditions, it is possible for R1 to deadlock.

Workaround

On r0p0, there is no workaround.

On r1p0, this erratum can be avoided by setting bit[27] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<27
MSR S3_0_C15_C1_1, x0
```

Setting this bit is not expected to have any visible performance impact.

2420947
Executing a store and store release on the same cacheline might lead to corruption

Status

Fault Type: Programmer Category B.
Fault Status: Present in r1p0. Fixed in r1p1.

Description

Executing a store and a store release accessing the same cacheline, but on different halves, might lead to data corruption.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- A store instruction W1 is executed on cacheline A, on a 32-byte half.
- A store release instruction W2 is executed on cacheline A, but on the other 32-byte half.

Implications

If the previous conditions are met and under very specific timing conditions, it is possible that W1 is being written twice in memory, leading to corruption.

Workaround

Only r1p0 is affected, the erratum can be avoided by setting bit[33] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<33
MSR S3_0_C15_C1_1, x0
```

This will prevent store and store-release to merge inside the write buffer, and it is not expected to have much performance impacts.

2418468
Crossing 4K loads might cause ordering violation or silent corruption in MTE asymmetric mode

Status

Fault Type: Programmer Category B.
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

Crossing 4K loads might cause:

- ordering violation (case 1): Read-after-Read (RaR) ordering is violated on any bytes inside the first page
- silent corruption (case2): if an SVE load is performed, First Fault Register (FFR) is not reported and it is causing a silent corruption

Configurations Affected

This erratum affects all configurations when BROADCASTMTE=1.

Conditions

Case 1 of the erratum occurs if memory tagging is enabled in asymmetric mode (SCTLR_ELx.TCF = 0b11) and if the following conditions are met:

- Core0 executes a store at address [0x40]
- Core1 executes a store at address [0x3F]
- Core0 executes a load0 at address [0x3F-0x40]
- Core0 executes a load1 at address [0x3F]

Case 2 of the erratum occurs if memory tagging is enabled in asymmetric mode (SCTLR_ELx.TCF = 0b11) and if the following conditions are met:

- Core0 executes a store at address [0x40]
- Core0 executes an SVE load at address [0x3F-0x40], and:
  - If First-fault: observe a watchpoint, ECC, or MTE faults on the first page and not on the first active element.
  - If Non-fault: observe any fault on the first page.

Implications

If the previous conditions are met and under very specific timing conditions, it is possible that:

- Case 1: "Core0 load0" observes "Core1 store" while "Core0 load1" doesn't, causing an RaR violation.
- Case 2: The fault is not reported into the FFR register, causing a silent corruption.

Workaround

This erratum can be avoided by not using memory tagging in asymmetric mode.

2413290
Enabling the Statistical Profiling Extension might lead to deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

Description

When the *Statistical Profiling Extension* (SPE) is enabled, selecting specific operations for sampling might cause the PE to deadlock.

Configurations Affected

This erratum affects configurations with SPE=TRUE.

Conditions

This erratum can occur when all of the following conditions are met:

- The Statistical Profiling Extension is enabled
- A NEON or SVE Load instruction is selected for sampling and the selected instruction:
  - accesses Non-cacheable memory
  - is architecturally executed

Implications

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

Workaround

This erratum can be avoided by setting bit[58:57]='b11 in CPUACTLR_EL1:

```
MRS x0, S3_0_C15_C1_0
ORR x0, x0, #3<<57
MSR S3_0_C15_C1_0, x0
```

2409570
Store-Release instructions might not be observed in the correct order

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, and r1p0. Fixed in r1p1.

Description

Two stores at the same address can be made visible in the incorrect order, violating Write-After-Write ordering.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- A store-release instruction W1 is executed at address A.
- A store instruction W2 is executed at address A.
- A store instruction W3 is executed at address A.

Implications

If the previous conditions are met and under very specific timing conditions, it is possible that W3 is made visible before W2, leading to incorrect data being observed by subsequent loads at address A. This behavior is not transient, and the final memory value is the one written by W2.

Workaround

On r0p0, this erratum can be avoided by inserting a DMB ST before each store-release using the following sequence

```
MOV x0, #0
MSR s3_6_c15_c8_0, x0
ISB
LDR x0, =0x0880_8000
MSR s3_6_c15_c8_2, x0
LDR x0, =3fe0_8000
MSR s3_6_c15_c8_3, x0
MOV x1, #0
ORR x1, #1<<0
ORR x1, #3<<4
ORR x1, #0xf<<6
```

```
ORR x1, #1<<20
ORR x1, #1<<33
MSR s3_6_c15_c8_1, x1
MOV x0,#1
MSR s3_6_c15_c8_0, x0
ISB
LDR x0, =0x1900_0000
MSR s3_6_c15_c8_2, x0
LDR x0, =3fe0_0c00
MSR s3_6_c15_c8_3, x0
MSR s3_6_c15_c8_1, x1
ISB
```

On r1p0, this erratum can be avoided by setting bit[32] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<32
MSR S3_0_C15_C1_1, x0
```

The workaround is not expected to have a significant performance impact for software that makes occasional use of store-release instructions.
The workaround is expected to have a significant performance impact for software that relies heavily on using store-release instructions.

2376701
LDFF/LDNF can generate a corrupted First Fault Register

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, and r1p0. Fixed in r1p1.

Description

LDFF and LDNF are *Scalable Vector Extension* (SVE) load instructions that store the abort position of memory access in the *First Fault Register* (FFR) register. Under some conditions, the FFR register might not be updated correctly, which means that its value is corrupted.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following conditions applies:

- The *Processing Element* (PE) executes a LDFF or a LDNF instruction, which causes a data-abort, requiring a FFR update.

Implications

When the above condition is met, under some timing conditions, the LDFF/LDNF instruction will not produce the correct value in the FFR register.

Workaround

The erratum can be avoided by setting bit[0] in CPUACTLR_EL1:

```
MRS x0, S3_0_C15_C1_0
ORR x0, x0, #1<<0
MSR S3_0_C15_C1_0, x0
```

Setting this bit is not expected to have a significant impact on performance.

2344187
Slow MRS instruction can get corrupted data

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, and r1p0. Fixed in r1p1.

Description

Some MRS instructions have long access latencies, and for these instructions the core allows other instructions to speculatively execute using the same execution unit.
Under specific microarchitectural conditions, if an IRG, ERETAA, or ERETAB is speculatively executed while such an MRS instruction is executed, the MRS instruction might return an incorrect value.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- the PE executes an MRS instruction accessing one of the following registers:
  - any of the IMP_CLUSTER*_EL1 registers
  - IMP_CPUPWRCTLR_EL1
  - any of the ERX*_EL1 registers if ERRSELR=0
  - any generic timer registers
- no instruction reading the register written by MRS is executed before the instruction generating an exception
- an instruction in program order after this MRS generates an exception
- one of the following instructions executes speculatively, and in program order after the instruction causing the exception:
  - IRG with GCR_EL1.RRND=0
  - ERETAA, ERETAB with PAC enabled

Implications

When the previous conditions are met, the MRS instruction will produce data that does not match the value of the System register being accessed.

Workaround

On IRG instruction, this erratum can be avoided by setting GCR_EL1.RRND to 0b1.
On ERETAA or ERETAB instruction, this erratum can be avoided by using the following sequence:

```
MOV x0, #2
MSR s3_6_c15_c8_0, x0
ISB
LDR x0, =0xd69f_0bff
MSR s3_6_c15_c8_2, x0
LDR x0, =0xffff_fbff
MSR s3_6_c15_c8_3, x0
MOV x1, #0
ORR x1, #1<<0
ORR x1, #3<<4
ORR x1, #0xf<<6
ORR x1, #1<<20
ORR x1, #1<<53
MSR s3_6_c15_c8_1, x1
```

2331818
Store might never complete a coherency operation, leading to deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

Under very specific timing conditions, a store might never complete a coherency operation, leading to deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- The core starts a coherency operation in order to perform a store at address A.
- The core starts multiple other coherency operations in order to perform other stores at different addresses.

Implications

If the conditions are met, under very specific microarchitectural timing conditions, the PE can deadlock.

Workaround

This erratum can be avoided by setting bit[20] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<20
MSR S3_0_C15_C1_1, x0
```

Setting this bit is expected to have a negligible performance impact

2330952
A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

A continuous stream of incoming *Distributed Virtual Memory* (DVM) syncs might cause the *Trace Buffer Extension* (TRBE) to prevent the core from forward progressing, while executing a WFx.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- The *Processing Element* (PE) executes a WFE or WFI instruction.
- TRBE is in use and needs to write trace data to its buffer.
- A continuous stream of DVM sync operations is received from other PEs.

Implications

When all of the above conditions are met, the PE might be prevented from entering WFE or WFI, and the pending WFE or WFI operation cannot be interrupted.

Workaround

There is no workaround.

2317812
Sequential consistency might not be respected with a non cacheable Store-Release

Status

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

Description

A Load Acquire instruction might be observed before an older Store Release instruction using a memory type other than Normal Cacheable, Inner Write-Back, Outer Write-Back.

Configurations Affected

This erratum only affects configurations using Theodul version r3p1 or later releases.

Conditions

The erratum occurs if all the following conditions are met:

- A Store-Release instruction W1 is executed at an address whose memory attributes are any of the following:
  - Device memory
  - Normal memory Non-Cacheable
  - Cacheable Write-Through
  - Cacheable Non-Shared

- An `LDARB`, `LDARH`, `LDAR`, `LDLARB`, `LDLARH`, or `LDLAR` instruction R2 is executed.
- Some rare, timing-dependent micro-architectural conditions occur.

Implications

If the above conditions are met, it is possible for R2 to be observed before W1.

Workaround

Typical software would generally not use `STLR`/`LDAR` to create and enforce ordering semantics for the affected memory types. If software does rely on `STLR`/`LDAR` to create ordering for memory types other than Normal Cacheable, Inner Write-Back, Outer Write-Back, explicit barriers can be used to enforce the expected ordering.

2302347
Write-After-Write ordering might be violated in presence of STLR

Status

Fault Type: Programmer CatB
Fault Status: Present in r0p0. Fixed in r1p0.

Description

Two stores at the same address can be made visible in the incorrect order, violating Write-After-Write ordering.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- A store instruction W1 is executed at address A.
- A store release instruction is executed at address A.
  - This instruction is not part of the architectural execution of the program and is later discarded.
- A store instruction W2 is executed at address A.

Implications

If the above conditions are met, it is possible that W2 is made visible before W1, leading to incorrect data being observed by subsequent loads at address A. This behavior is not transient, and the final memory value is the one written by W1.

Workaround

This erratum can be avoided by injecting a `DMB ST` before each Store Release, using the following sequence:

```
MOV x0, #0
MSR s3_6_c15_c8_0, x0
ISB
LDR x0, =0x0880_8000
MSR s3_6_c15_c8_2, x0
LDR x0, =3fe0_8000
MSR s3_6_c15_c8_3, x0
MOV x1, #0
ORR x1, #1<<0
```

```
ORR x1, #3<<4
ORR x1, #0xf<<6
ORR x1, #1<<20
ORR x1, #1<<33
MSR s3_6_c15_c8_1, x1
MOV x0,#1
MSR s3_6_c15_c8_0, x0
ISB
LDR x0, =0x1900_0000
MSR s3_6_c15_c8_2, x0
LDR x0, =3fe0_0c00
MSR s3_6_c15_c8_3, x0
MSR s3_6_c15_c8_1, x1
ISB
```

2292761
Executing code in a context with multiple VAs mapped to the same PA might lead to an opcode corruption

Status

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

Description

Executing code in a context with multiple VAs mapped to the same PA might result in execution of opcodes different from the ones at the physical location.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum can happen if:

1. Multiple VAs are mapped to the same PA within the same context and;
2. At least two of the VA mappings have execution permissions.

Implications

This erratum might lead to an opcode corruption.

Workaround

This erratum can be avoided by setting CPUACTLR4_EL1[13] to 1

```
MRS x0, S3_0_C15_C1_3
ORR x0,x0,#0x2000
MSR S3_0_C15_C1_3, x0
```

Note that using this workaround has no performance impact.

2287512
Entry into the Full Retention power mode might cause corruption on Itag, Idata, and TLB RAMs

Status

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

Description

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

Configurations Affected

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

Conditions

The erratum occurs if all the following conditions apply:

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

Implications

If the conditions are met, the RAM contents of the itag, idata, and L2 TLB RAMs might be corrupted. As a result, the PE might:

- fetch and execute incorrect opcodes as a result of itag/idata corruption
- use incorrect translation information as a result of TLB RAM corruption

Workaround

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

2285473
A LDR executed between two Non-Cacheable LDRs to the same 64B boundary might fail to apply ordered-before semantics to said instructions

Status

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

Description

A load instruction (`LDR`) might fail to apply ordered-before semantics to a subsequent load instruction to a Non-Cacheable location.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

- A Load instruction R1 is executed.
- A Load instruction R2 is executed.
- A Load instruction R3 is executed at an address within the same 64B, 64B-aligned granule as R1.
- R2 is ordered-before R3.
- R1 and R3 access a page whose memory attributes are any of:
  - Normal memory Non-Cacheable
  - Cacheable Write-Through
  - Cacheable Non-Shared

Implications

If the above conditions are met, the PE might fail to maintain ordered-before semantics for R2 and might observe R3 before R2.

Workaround

This erratum can be avoided by setting bit[12] in CPUACTLR3_EL1:

```
MRS x0, S3_0_C15_C1_2
ORR x0, x0, #1<<12
MSR S3_0_C15_C1_2, x0
```

2285536
TRBE might generate corrupt trace data and/or fail to complete all pending accesses following a DSB or DVM Sync

Status

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

Description

Executing a DSB instruction or receiving a DVM Sync operation might cause TRBE to either corrupt the trace or continue using an invalidated mapping after the DVM Sync completion.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- TRBE is enabled and generating trace data
- a DVM Sync operation is received from a different PE, or a DSB instruction is executed on the PE

Implications

When the conditions are met, TRBE can do one of the following:

1. continue writing to the buffer after the completion of the DSB or DVM Sync (even if the DSB or DVM Sync should have ensured completion of all writes to the buffer). This can only happen if the software controlling TRBE updates the TRBE MMU mappings while TRBE is in use.
2. modify the value of TRBPTR_EL1 to a previously-held value between TRBBASER_EL1 and TRBLIMITR_EL1, and write a trace packet at this address

If the TRBE MMU mappings are not updated dynamically, the only implication of this erratum is item 2, which might cause part of the trace date to be corrupted.

Workaround

This erratum can be avoided by disabling TRBE.

2284544
Exclusive store might deadlock under very specific timing conditions

Status

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

Description

Under very specific timing conditions, an exclusive store might never complete, leading to deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- The core starts, speculatively, a coherency operation in order to perform a store at address A.
- The core executes an older exclusive load at address A.
- The core executes an older exclusive store at address A.

Implications

If the conditions are met, under very specific microarchitectural timing conditions, the PE can deadlock.

Workaround

This erratum can be avoided by setting bit[20] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<20
MSR S3_0_C15_C1_1, x0
```

2275754
Speculatively executed store behind constantly mis-predicted branch might prevent forward progress of synchronization primitives in other PEs

Status

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

Description

Under certain conditions, a loop might continuously mis-predict. If the speculative instruction path has a store instruction to the same physical address as another PE's exclusive monitor address, then this might cause a repeatable loop where the cache line is requested by the store instruction to be unique, opening the exclusive monitor on the other PE.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- There is a loop that has a branch that is consistently mis-predicted.
- There is a store instruction outside of the loop that has the same physical address as the exclusive monitor address of another PE, within the cache line boundary. The store instruction makes a unique request, snooping that cache line from other PEs, and opening the exclusive monitor.
- The synchronization primitive used by the victim PE is an LDXR-STXR pair.

Implications

If the previous conditions are met, the PE performing the synchronization primitive might livelock.

Workaround

This erratum can be avoided by setting bit[20] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<20
MSR S3_0_C15_C1_1, x0
```

2268070
Data Cache Clean operations by VA to the Point of Coherency, Point of Persistence, or Point of Deep Persistence might cause the PE to deadlock

Status

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

Description

Under specific timing conditions, the execution of a Data Cache Clean operation by VA to the Point of Coherency, Point of Persistence, or Point of Deep Persistence might cause the PE to deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

- The core executes one of the following Data Cache Clean (not Invalidating) operations by VA to the Point of Coherency, Persistence, or Deep Persistence:
  - DC CGDVAC
  - DC CGDVADP
  - DC CGDVAP
  - DC CGVAC
  - DC CGVADP
  - DC CGVAP
  - DC CVAC
  - DC CVADP
  - DC CVAP
- The line is present in Clean state in the L2 cache
- A snoop to the same line is received before the Clean operation completes

Implications

If the previous conditions are met, under specific micro-architectural and timing conditions, the PE can deadlock.

Workaround

- For Clean operations to the Point of Coherency, set bit[10] of CPUACTLR2_EL1, which changes them to Clean and Invalidate operations.
- There is no workaround for Clean operations to the Point of Persistence or Deep Persistence. These are not expected to be used on sample silicon.

2248167
ESR_ELx and FAR_ELx might report incorrect information on Instruction Abort when hardware breakpoints are enabled

Status

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

Description

ESR_ELx and FAR_ELx might report incorrect information on Instruction Abort when hardware breakpoints are enabled.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- Hardware breakpoints are enabled.
- A hardware breakpoint is triggered, either speculatively or as part of the architecturally executed sequence of instructions.

Implications

- When a hardware breakpoint is triggered as part of the architecturally executed path, the ESR_ELx and FAR_ELx might report information consistent with an Instruction Abort that has been taken on a different instruction or for a different reason, rather than a Breakpoint Exception
  - In this situation, the resulting fault is routed consistently with a Breakpoint Exception being taken.
- When an Instruction Abort is detected as part of the architecturally executed path, the ESR_ELx and FAR_ELx might report information consistent with a Breakpoint Exception that has occurred on a different instruction.
  - In this situation, the resulting fault is routed consistently with the information presented in ESR_ELx.

Workaround

This erratum can be avoided by using software breakpoints instead of hardware breakpoints.

2239006
The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation.

Status

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

Description

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

Configurations Affected

All configurations are affected.

Conditions

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

Implications

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

Workaround

The erratum can be avoided by setting bit[9] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<9
MSR S3_0_C15_C1_1, x0
```

Setting this bit is not expected to have any significant performance impact for test silicon.

2238661
In MTE precise mode, Load with writeback can cause the PE to deadlock

Status

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

Description

Under some timing conditions, a load with writeback can cause the PE to deadlock.

Configurations affected

All configurations are affected

Conditions

This erratum occurs under the following conditions:

- MTE is enabled in precise checking mode.
- A load with write-back is executed.
- The load instruction causes a micro-architectural hazard requiring the following instructions to be replayed.

Implications

If the above conditions are met and under specific timing conditions, the load instruction can cause the PE to deadlock.

Workaround

The erratum can be avoided by setting bit[0] in CPUACTLR_EL1:

```
MRS x0, S3_0_C15_C1_0
ORR x0, x0, #1<<0
MSR S3_0_C15_C1_0, x0
```

Setting this bit is not expected to have a significant performance impact.

2180026
Reset value of MDCR_EL2.HPMN does not reflect the number of implemented PMU counters

Status

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

Description

The reset value of MDCR_EL2.HPMN is architecturally defined to be the value of PMCR_EL0.N, reflecting the number of available PMU counters.

Due to this erratum, the reset value of MDCR_EL2.HPMN is always 6 regardless of the implemented number of PMU counters.

Configurations affected

All configurations with PMU_CNT_NUM=20 are affected. Other configurations are not affected.

Conditions

There is no specific set of conditions required to trigger this erratum.

Implications

Software relying on the reset value of MDCR_EL2.HPMN to reflect the number of implemented counters might be limited to using the first 6 counters instead of the total number of available counters.

Workaround

Software can set MDCR_EL2.HPMN to the maximum implemented number of PMU counters using the following sequence:

```
MRS X0, PMCR_EL0
UBFX X0, X0, #11, #5

MRS X1, MDCR_EL2
AND X1, X1, #0xfffffffe0

ORR X1, X1, X0
MSR X1, MDCR_EL2
```

Category B (rare)

There are no errata in this category.

Category C

2731260
Some PMU events might have incorrect values

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

Description

Some *Performance Monitoring Unit* (PMU) events might be counted incorrectly and either be incremented in situations where they should not, or fail to be incremented in situations where they should.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if a given PMU counter is configured to count the following event:

- 0x0029, L3D_CACHE_ALLOCATE

The erratum occurs if a given PMU counter is configured to count one of the following events under the listed situations:

- 0x0074, ASE_SPEC
- 0x0075, VFP_SPEC
- 0x0077, CRYPTO_SPEC
- In the following situations:
  - Advanced SIMD data-processing instructions might incorrectly be counted as VFP_SPEC instead of ASE_SPEC.
  - PMUL instructions from the 'Advanced SIMD three same' instruction class might be counted as CRYPTO_SPEC instead of ASE_SPEC.

The erratum occurs if a given PMU counter is configured to count one of the following events under the listed situations:

- 0x80E7, ASE_SVE_INT16_SPEC
- 0x80EB, ASE_SVE_INT32_SPEC
- 0x80EF, ASE_SVE_INT64_SPEC
- In the following situation:

- PMUL instructions from the 'Advanced SIMD three same' instruction class might fail to be counted.

Implications

If the previous conditions are met, the affected PMU events will be incorrect.

Workaround

This erratum has no workaround.

2712353
Information in some Statistical Profiling Extension packets might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0, r1p1 and r1p2. Open.

Description

Some information contained in the *Statistical Profiling Extension* (SPE) packets might be incorrect.

Configurations Affected

This erratum affects configurations with SPE=TRUE.

Conditions

No specific conditions are required to hit this erratum.

Implications

The Operation Type packet field CLASS[1:0] might be incorrect if the tracked operation is one of:

- LDRAA instruction
- LDRAB instruction

The instructions are incorrectly assigned a type of 'Other', and the information presented in the corresponding SPE packet is consistent with a packet of type 'Other'.

This also affects filtering of the packets whenever filtering is configured to use some of the affected packet contents.

Workaround

This erratum has no workaround.

2709709
Interrupts might be taken following a Context Synchronization Event when MDSR_EL1.INTdis or EDSCR.INTdis are set

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

When an interrupt is pending when executing a *Context Synchronization Event* (CSE) that synchronizes setting MDSR_EL1.INTdis or EDSCR.INTdis to 1, the interrupt might be taken immediately after the CSE and before any subsequent instructions, even though interrupts should be disabled.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions are met:

- ExternalInvasiveDebugEnabled() or ExternalSecureInvasiveDebugEnabled() is TRUE
- One of the following conditions is met:
  - the PE executes an MSR MDSCR_EL1 setting INTdis to 1
  - an external agent sets EDSCR.INTdis to 1
- An interrupt is pending when the CSE synchronizing the change to INTDis is performed.

Implications

The interrupt is taken immediately after the CSE and before executing any subsequent instruction.

Workaround

- When INTdis is set using MSR MDSCR_EL1, no workaround is required.
- When INTdis is set using a write to EDSCR, this erratum can be avoided by executing a CSE before existing Debug State.

2701325
The PE might not respond to APB accesses to some registers in OFF_EMU

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0 and r1p1. Fixed in r1p2.

Description

The PE might not respond to APB accesses to some registers in OFF_EMU

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all of the following conditions are met

- The PE is in OFF_EMU state
- An APB access to one of the following registers:
  - DBGBVR<n>
  - DBGBCR<n>
  - DBGWVR<n>
  - DBGWCR<n>
  - EDWAR
  - EDITR
  - EDECCR

Implications

When the above conditions are met, the PE stop responding to APB traffic while it remains in OFF_EMU state.

Workaround

This erratum has no workaround.

2661229
ESR_ELx for a Breakpoint exception might be incorrect if breakpoints are concurrently disabled through APB

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum happens when all of the following conditions are met:

- The PE architecturally executes an ERET instruction that causes PSTATE.IL to be set to 1
- A breakpoint was set to hit on the target address of the ERET instruction
- An external debugger concurrently disables this breakpoint through the APB interface

Implications

If the above conditions are met:

- The PE generates an exception that will be routed consistently with a Breakpoint exception
- The value of ESR_ELx for this exception is consistent with an Illegal Execution state exception
- ELR_ELx and FAR_ELx correctly point to the instruction that generated the exception

Workaround

This erratum has no workaround.

2631695
Incorrect transition to Software-Step Active state after OSLAR_EL1.OSLK write followed by ERET

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

When executing an ERET instruction, synchronization of pending changes to OSLAR_EL1.OSLK does not take effect correctly, resulting in a transition of Software-Step state-machine to Active state.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all of the following conditions are met:

- The *Processing Element* (PE) is executing at the Exception Level ELx
- The software-step is enabled at a lower exception level, ELy
- The following two instructions are executed:
  - MSR OSLAR_EL1that sets the OSLK bit to 1
    - this disables software-step at ELy
  - ERET to ELy

Implications

If the above conditions are met, Software-Step will incorrectly transition to Active state instead of staying in an Inactive state. The Software-Step exception can happen immediately after ERET, or after one stepped instruction, depending on the value of SPSR_ELx.SS before ERET.

Workaround

This erratum can be avoided by not relying on ERET to synchronize the change to OSLAR_EL1.OSLK, and to use an explicit ISB instruction instead. It is expected that most operating systems already do this in practice.

2631722
RAMINDEX operations accessing the L1 Data cache might fail to return the correct result

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

RAMINDEX operations accessing the L1 Data cache might fail to return the correct result.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- A RAMINDEX operation targeting the L1 Data cache (Xt[31:24]=0x9).
- The operation targets an address with VA[5:4]='b11.

Implications

If the previous conditions are met, the RAMINDEX operation will fail to populate the operation result registers (IMP_DSIDE_DATA0_EL3 and IMP_DSIDE_DATA1_EL3) with the value from the targeted RAM location.

Workaround

This erratum has no workaround.

2625935
UNDEFINED exceptions due to EDSCR.SDD might be prioritized over EL2 traps caused by HCR_EL2.TIDCP

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

For MSR instructions to IMPLEMENTATION DEFINED registers which have access controlled by an EL3 register, UNDEFINED exceptions due to EDSCR.SDD are incorrectly prioritized over EL2 traps caused by HCR_EL2.TIDCP.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- the PE is in the following state:
  - PSTATE.EL == EL1
  - HCR_EL2.TIDCP == 1
  - EDSCR.SDD == 1
  - Halted() returns TRUE
- The *Processing Element* (PE) executes an MSR instruction to one of the following registers:
  - IMP_CPUACTLR2_EL1 and ACTLR_EL3.ACTLREN == 0
  - IMP_CPUACTLR3_EL1 and ACTLR_EL3.ACTLREN == 0
  - IMP_CPUACTLR4_EL1 and ACTLR_EL3.ACTLREN == 0
  - IMP_CPUACTLR_EL1 and ACTLR_EL3.ACTLREN == 0
  - IMP_CPUECTLR2_EL1 and ACTLR_EL3.ECTLREN == 0
  - IMP_CPUECTLR_EL1 and ACTLR_EL3.ECTLREN == 0
  - IMP_CPUPPMPDPCR_EL1 and ACTLR_EL3.PDPEN == 0
  - IMP_CPUPWRCTLR_EL1 and ACTLR_EL3.PWREN == 0

Implications

When the previous conditions are met, the UNDEFINED exception caused by EDSCR.SDD will be prioritized over the EL2 trap caused by HCR_EL2.TIDCP.

Workaround

This erratum has no workaround.

2574284
Software-step with no instruction step after debug-exit from EL0 with a bad mode in DSPSR

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

On exit from Debug state, PSTATE.SS is set according to DSPSR.SS and DSPSR.M.
If DSPSR.M encodes an illegal value, then PSTATE.SS should be set according to the current Exception level. When the erratum occurs, the *Processing Element* (PE) always writes PSTATE.SS to 0.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- DSPSR.M encodes an illegal value.
- DSPSR.SS is set.
- The PE exits Debug State while in EL0 with Software Step enabled.

Implications

If the previous conditions are met, then, on exit from Debug state the PE will directly take a Software-step Exception, instead of stepping an instruction as would be expected from DSPSR.SS=1.

Workaround

This erratum has no workaround.

2460643
Checked stores in precise mode might fail to report tag check fails

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open

Description

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

Configurations Affected

This erratum affects configurations with BROADCASTMTE=1.

Conditions

The erratum occurs when all the following conditions are met:

- MTE is enabled in precise checking mode (SCTLR_ELx.TCF='b01).
- An instruction R1 with acquire semantics is executed. R1 can be one of:
  - Any LDAR* or LDAP* instruction.
  - Any SWP*, CAS* or LD* Atomic instruction with acquire semantics.
- A tag-checked store instruction W3, performing a Tag-Check-read R2, is executed.
- R1 is in program order before W3.

Implications

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

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

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

Workaround

This erratum has no workaround.

2455253
Crossing 4KB load might report incorrect FFR index

Status

Fault Type: Programmer Category C.
Fault Status: Present in r1p0, r1p1 and r1p2. Open.

Description

Executing a crossing 4KB SVE Non-Fault or First-Fault load might lead to an incorrect value being reported to the FFR register in case on a watchpoint fault and a tag_check_fail or an *Error Correcting Code* (ECC) fault that are being observed after the 4KB boundary.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- The *Processing Element* (PE) executes a 4KB-crossing SVE Non-Fault or First-Fault contiguous load.
- The load observes a watchpoint fault on a byte higher than the first one after the 4KB boundary.
- The load observes a MTE tag_check_fail or an ECC double bit error on a byte lower than the watchpoint, after the 4KB boundary.

Implications

If the above conditions are met, and under specific timing conditions, it is possible for the FFR to be updated with the position of the watchpoint fault instead of the tag_check_fail or ECC fault.

Workaround

There is no workaround.

2428886
ERXSTATUS_EL1.OF might incorrectly be set in case of multiple errors

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, and r0p1. Fixed in r1p1.

Description

On a RAS report, ERR1STATUS.OF is supposed to be set when the error record has overflowed with multiple errors, or if at least one error syndrome has been discarded.

If the PE detects multiple deferred or uncorrected errors, the PE sets ERR1STATUS.OF to 1, which is correct.

If the PE detects multiple corrected errors, the PE will set the ERR1STATUS.OF to 1, which is incorrect.

The PE might also set ERR1STATUS.OF to 1 when two different types of errors are detected at the same time, which is incorrect.

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION=1.

Conditions

Multiple ECC errors are detected by the PE at the same time.

Implications

In the presence of multiple errors, the reporting of overflow can be inaccurate, which might lead to error handling software overestimating or underestimating the number of errors that have occurred. This might result in a negligible increase in the failure in time (FIT) rate.

Workaround

This erratum has no workaround.

2426380
Cacheable load might return corrupted data upon single-bit error in L1 data cache tag RAM

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

When a single-bit ECC error happens on a load instruction, it is expected to be detected and corrected. When this erratum happens, it might fail to be detected, causing the load to return corrupted data to the core.

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION=1.

Conditions

The erratum can occur when two loads, R1 and R2, are executed at the same time.

Additionally, all of the following conditions must apply:

- R1 and R2 are executed at the same virtual address.
- R1 and R2 are executed at the same physical address.
- R1 and R2 load the same number of bytes from memory.
- R1 is unpredicated or has at least one active element.
- R2 is predicated and has zero active elements.
- A single-bit ECC error on the L1 data cache tag RAM is present during the execution of R1 and R2

Implications

When the erratum happens, incorrect data might be returned by R1 and consumed by the PE. The presence of the ECC error is still correctly logged in the RAS registers.

Workaround

This erratum has no workaround.

2424427
Uncacheable 32-bytes loads might fail to report poison information

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

Description

When poisoned data is received during the execution of an uncacheable load, the error should be reported through an SError interrupt. When this erratum happens, this might fail to be reported.

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION=1.

Conditions

The erratum can occur when a 32-bytes-wide load instruction is executed.

The affected load uses one of the following memory types:

- Cacheable Write-Through
- Cacheable non-shared
- Non-cacheable
- Device

Implications

When the erratum happens, poison information might fail to be reported by the affected instruction. In such a case, incorrect data are returned and consumed by the PE.

Workaround

This erratum has no workaround.

2422640
A continuous flow of snoops and atomics might prevent a store from becoming visible in finite time

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

A continuous flow of snoops from other cores combined with a continuous flow of atomics might prevent an older store from becoming visible in finite time.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under all of the following conditions:

- Core0 executes a store to cacheable address A.
- Core1 executes a continuous flow of loads or stores at address A.
- Core0 executes a continuous flow of atomic operations to another address B.

Implications

If the above conditions are met, the store operation executed on the Core0 might not be visible in finite time. The core itself keeps performing forward progress and stays interruptible.

Workaround

No workaround is deemed necessary for this erratum.

2411188
ESR_ELx.ISV might be incorrect when software-stepping an MSR DAIF instruction

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

When stepping an MSR DAIF instruction, ESR_ELx.ISV might incorrectly be set to 0 instead of 1 even if the instruction has stepped.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs if all the following conditions apply:

- The Software-Step state machine transitions to active-not-pending state.
- The stepped instruction is an MSR DAIF that causes PSTATE.D to transition from 1 to 0.

Implications

If the above conditions are met, ESR_ELx.ISV will be set to 0 instead of 1.

Workaround

There is no workaround.

2389413
Some PMU events might have incorrect values

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

Some *Performance Monitoring Unit* (PMU) events might be counted incorrectly and either be incremented in situations where they should not, or fail to be incremented in situations where they should.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if a given PMU counter is configured to count one of the following events:

- 0x002B, L3D_CACHE
- 0x0073, DP_SPEC
- 0x0074, ASE_SPEC
- 0x0075, VFP_SPEC
- 0x00A0, L3D_CACHE_RD
- 0x8006, SVE_INST_SPEC
- 0x8018, FP_HP_SPEC
- 0x8018, FP_SP_SPEC
- 0x801C, FP_DP_SPEC
- 0x80E3, ASE_SVE_INT8_SPEC
- 0x80E7, ASE_SVE_INT16_SPEC
- 0x80EB, ASE_SVE_INT32_SPEC
- 0x80EF, ASE_SVE_INT64_SPEC

Implications

If the previous conditions are met, the affected PMU events will be incorrect.

Workaround

This erratum has no workaround.

2387304
Checked crossing 64B boundary load might fail to report a tag check fault

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

A checked load crossing the 64B boundary might ignore a tag check fail on the high 16B, leading to synchronous exception not being taken (SCTLR_ELx.TCF=0b01), or TFSR_ELx not being updated (SCTLR_ELx.TCF=0b10).

Configurations Affected

This erratum affects configurations with BROADCASTMTE=1.

Conditions

The *Processing Element* (PE) is running with Memory Tagging enabled in Synchronous or Asynchronous Mode (SCTLR_ELx.TCF=0b01 or 0b10).

Additionally, the PE is executing the following instructions:

- Checked load crossing 64B boundary

Implications

If the previous conditions are met, and under very specific timing conditions, the checked load might not report the Tag Check Fail result on range [0x40-0x4F].

Workaround

This erratum has no workaround.

2385318
PMCFGR.EX and PMCR_EL0.X should be 0

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

Description

PMCFGR.EX and PMCR_EL0.X respectively expose support and control of the PMU event export bus. This feature is not supported in this product and the bits should consequently both be RES0.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum happens when software reads the value of PMCFGR.EX.

Implications

A software reading PMCFGR.EX to determine whether a PMU event export bus is supported will incorrectly think that the feature is supported because PMCFGR.EX='0b1'.

Workaround

No workaround is required for this erratum.

2372608
Double bit error in L2 Tag RAM might lead to deadlock or protocol violation

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

A transient double-bit *Error Correction Code* (ECC) error in the L2 Tag RAM might result in a line being present twice inside of the core. This might lead to a deadlock or coherency protocol violation.

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION=True.

Conditions

Under specific microarchitectural and timing conditions, a transient L2 double-bit ECC error in the L2 Tag RAMs might lead to a cache line being present twice inside the core.

Implications

If the previous conditions are met, the Processing Element can eventually deadlock or violate the coherency protocol.

Workaround

This erratum has no workaround.

2354488
Information in some Statistical Profiling Extension packets might be incorrect

Status

Fault Type: Programmer Category C.
Fault Status: Present in r1p0. Fixed in r1p1.

Description

Some information contained in the *Statistical Profiling Extension* (SPE) packets might be incorrect.

Configurations Affected

This erratum affects configurations with SPE=TRUE.

Conditions

No specific conditions are required to hit this erratum.

Implications

- The Events packet field E[11], 'Alignment', might be wrong.
- The Events packet field E[9], 'Last level cache miss', might be wrong in a multi-cluster system, or if the tracked operation is one of:
  - An Atomic memory operation, LD<op> or ST<op>
  - A Swap instruction
  - A Compare and Swap instruction
- The Events packet field E[8], 'Last level cache access', might be wrong.
- The Operation Type packet field SUBCLASS.FP might be wrong for the following SVE instructions:
  - SDIV, UDIV, SDIVR, USDIVR, URECPE and URSQRTE
- The Operation Type packet field SUBCLASS.P might be wrong for the following SVE instructions:
  - CLASTA, CLASTB
  - MOV (predicate, predicated, merging), MOV (vector, predicated)
  - SEL (predicates), SEL (vectors)
  - SPLICE
  - UXTB, UXTH, UXTW
- When the Operation Type packet CLASS fields has the value of 'b01 (Load, Store or Atomic), all other packets for the sampled operation might contain incorrect information.
- The Timestamp packet might contain a timestamp value that was selected prior to the time at which the sampled operation is selected.

This also affects filtering of the packets whenever filtering is configured to use some of the affected packet contents.

Workaround

The IMPLEMENTATION DEFINED 'Data Source' packet can be used to determine the origin of the data, and to compute whether the last-level cache was accessed and whether the access missed.

2328270
Trace Buffer Extension might fail to complete all pending accesses following a Data Synchronization Barrier or Distributed Virtual Memory Sync

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

Description

Executing a *Data Synchronization Barrier* (DSB) instruction or receiving a *Distributed Virtual Memory* (DVM) Sync operation might cause *Trace Buffer Extension* (TRBE) to continue using an invalidated mapping after the DVM Sync completion.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

- TRBE is enabled and is generating trace data.
- A DVM Sync operation is received from a different *Processing Element* (PE), or a DSB instruction is executed on the PE.

Implications

When the conditions are met, TRBE can continue writing to the buffer after the completion of the DSB or DVM Sync (even if the DSB or DVM Sync should have ensured completion of all writes to the buffer). This can only happen if the software controlling TRBE invalidates an existing TRBE Memory Management Unit mapping while TRBE is in use.

Workaround

This erratum can be avoided by ensuring that the pages used by TRBE are pinned by software at stage1, and at stage2 when enabled, so that they do not get unmapped while TRBE is running.

2318005
SPE Issue Latency counter might be incorrect on sampled operations generating an exception

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0, r1p1 and r1p2. Open.

Description

In some conditions, the issue latency counter value might contain an incorrect value. The problem occurs only on specific operations that generate an exception.

Configurations Affected

This erratum affects configurations with SPE enabled.

Conditions

The erratum occurs if a sampled operation generates an architectural exception.

Implications

The Issue Latency counter value in the SPE record is incorrect.

Workaround

Software can avoid this erratum by ignoring the reported value of the Issue Latency Counter when the Events packet indicates that the sampled operation generated an exception.

2312234
Set/Way CMO instructions might not affect a cache line stored in the L1 victim cache

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

The PE might keep some valid cache lines even after a full sweep of L1 + L2 caches with cache maintenance by set/way instructions.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under the following conditions:

- The PE executes a series of cache maintenance by set/way instructions that target the entire L1 and L2 caches.
- Some specific micro-architectural conditions occur.

Implications

When this erratum occurs, some cache lines might stay valid in the L2 cache, despite a sequence of cache maintenance by set/way instructions covering all the L2 sets and ways. Cleaning/invalidation of such lines is not guaranteed, but coherency is preserved.

Hardware flushing of caches is not affected by this erratum.

Workaround

For r0p0, this erratum has no workaround.

For r1p0, the erratum can be avoided by setting bit[40] in CPUECTLR_EL1 just before the stream of CMOs, and clear this bit at the end of the sequence. This can be useful for EL3 firmware that wants to perform manual flushing of L1+L2 caches.
Example of code sequence:

```
MRS x0, S3_0_C15_C1_4
```

```
ORR x0, x0, #1<<40
MSR S3_0_C15_C1_4, x0
ISB
...
CMO sequence
...
DSB
MRS x0, S3_0_C15_C1_4
BIC x0, x0, #1<<40
MSR S3_0_C15_C1_4, x0
ISB
```

Setting this bit is not expected to have any performance impact if enabled during the CMO sequence only.

2310079
PMBSR_EL1 might report a Data Abort instead of a Buffer Management Event when PMBPTR_EL1 reaches PMBLIMITR_EL1

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

Description

When PMBPTR_EL1 reaches PMBLIMITR_EL1, SPE should report a Buffer Management Event in PMBSR_EL1. When the erratum happens, PMBSR_EL1 might instead report a Data Abort exception.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

- The VA contained in PMBLIMITR_EL1 either:
  - would generate an Address Size Fault if stage1 translation is disabled for the owning translation regime
  - cannot be translated by TTRB0_ELx nor TTBR1_ELx if stage1 translation is enabled for the owning translation regime

Implications

If the above conditions are met, PMBSR_EL1 might report a Data Abort rather than Buffer Management Event.

Workaround

This erratum can be avoided by configuring PMBLIMITR_EL1 with a value that:

- would not generate an Address Size Fault if stage1 translation is disabled for the owning translation regime
- can be translated through either TTBR0 or TTBR1 if stage1 translation is enabled for the owning translation regime

2305209
Executing an ESB might fail to clear a masked Virtual SError Interrupt

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

ESB instruction is used to synchronize pending physical or virtual SError interrupts, and make sure that if they are masked, they are deferred to DISR_EL1/VDISR_EL2, and cleared. Due to this erratum, a virtual SError interrupt might not be cleared when it should be.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- HCR_EL2.VSE is set
- vSEI is enabled (HCR_EL2.TGE=0 & HCR_EL2.AMO=1)
- vSEI is masked due to either:
  - PSTATE.A=0
  - ExternalDebugInterruptsDisabled() returns TRUE
- An ESB instruction is executed in EL0 or EL1
- IRQ or FIQ is unmasked and becomes pending during execution of the ESB

Implications

On ESB execution, VDISR_EL2.A is set, but HCR_EL2.VSE is not cleared.

Workaround

This erratum has no workaround.

2301922
Some PMU events might have incorrect values

Status

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

Description

Some PMU events might be incorrectly counted and either be incremented in situations where they should not, or fail to be incremented in situations where they should.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if a given PMU counter is configured to count one of the following events:

- 0x0017, L2D_CACHE_REFILL
- 0x0052, L2D_CACHE_REFILL_RD
- 0x8075, SVE_PRED_EMPTY_SPEC
- 0x8076, SVE_PRED_FULL_SPEC
- 0x8077, SVE_PRED_PARTIAL_SPEC
- 0x8108, BR_IMMED_TAKEN_RETIRED

Implications

If the previous conditions are met, the PMU event counter will be incorrect.

Workaround

Do not use the previously described PMU events.

2293949
Instruction sampling bias exists in SPE implementation

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0, r1p1 and r1p2. Open.

Description

A PE that is used to perform instruction sampling using the SPE mechanism might exhibit sampling bias.

Configurations Affected

This erratum affects all configurations where SPE=1.

Conditions

1. SPE configured and utilized on the PE.

Implications

Software utilizing SPE might see unexpectedly high sample counts for some instructions and unexpectedly low sample counts for other instructions.

Workaround

This erratum has no workaround.

2276160
PMU event DTLB_WALK might incorrectly count some instructions in case of EPD/E0PD translation fault

Status

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

Description

The PMU event DTLB_WALK is not supposed to count cases where an instruction takes a translation fault because of TCR_ELx.EPDy or TCR_ELx.E0PDy or both bits. When this erratum occurs, such instructions might incorrectly increment the PMU counter.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- PMU is configured to sample event DTLB_WALK (0x0034).
- Stage 1 address translation is enabled through SCTLR_ELx.M bits.
- TCR_ELx.EPDy or TCR_ELx.E0PDy or both are configured to prevent addressing space access.
- An instruction is executed at a virtual address such that the configuration of the above bits causes it to take a level 0 translation fault.

Implications

If the previous conditions are met, the PMU event might be incorrect.

Workaround

This erratum has no workaround.

2268885
PE might generate spurious Branch Target exceptions when dynamically changing the GP bit in page tables

Status

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

Description

If a PE is executing a code from a given VA0:PA0 mapping with the GP bit set while another PE modifies the value of the page table entry for VA0 to PA1 with the GP bit unset, than the PE might subsequently consider PA1 to have the GP bit set and incorrectly report a BTI exception while executing from PA1.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

- PE0 executes instructions from a given mapping, VA0:PA0 with GP=1
- PE1 concurrently modifies the page table entry for VA0 to PA1, with GP=0

Implications

PE0 executing the code from PA1 can observe an unexpected Branch Target exception.

Workaround :

This erratum has no workaround.

2262629
Checked store in precise mode might deadlock in case of external abort

Status

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

Description

A checked store starting a coherency operation might never be able to complete tag check operation if the system return an external abort, leading to a deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

- MTE is enabled with store in precise mode (TCF=2'b01).
- The core started a coherency operation in order to perform a checked store.
- External abort is returned by the system.

Implications

If the conditions are met, under specific micro-architectural and timing conditions, the PE can deadlock.

Workaround

The kind of external abort that leads to this erratum is not expected to occur with Arm interconnect.

If this erratum occurs with a custom IP, it can be avoided by setting bit [20] in CPUACTLR2_EL1:

```
MRS x0, S3_0_C15_C1_1
ORR x0, x0, #1<<20
MSR S3_0_C15_C1_1, x0
```

2259398
Software-step not done after exit from Debug state with an illegal value in DSPSR

Status

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

Description

On exit from Debug state, PSTATE.SS is set according to DSPSR.SS and DSPSR.M.
If DSPSR.M encodes an illegal value, then PSTATE.SS should be set according to the current Exception level. When the erratum occurs, the PE always writes PSTATE.SS to 0.

Configurations Affected

This erratum affects all configurations.

Conditions

- Software-step is enabled in current Exception level
- DSPSR.M encodes an illegal value, like:
  - M[4] set
  - M is a higher Exception level than current Exception level
  - M targets EL2 or EL1, when they are not available
- DSPSR.D is not set
- DSPSR.SS is set

Implications

If the previous conditions are met, then, on exit from Debug state the PE will directly take a Software-step Exception, without stepping an instruction as expected from DSPSR.SS=1.

Workaround

This erratum has no workaround.

2252693
Table walks generated by the Trace Buffer Extension might use MPAM information of the current Exception level rather than the TBE owning translation regime

Status

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

Description

When the Trace Buffer Extension is enabled, the associated translation table walks should generate MPAM information associated with the Exception level owning the TBE translation. When the erratum occurs, the generated MPAM information is instead associated with the Exception level currently in use by the PE.

Configurations Affected

This erratum affects all configurations.

Conditions

- The Trace Buffer Extension is in use.
- TRBLIMITR_EL1.nVM == 0
- The owning translation regime for TBE is different from the translation regime currently in use by the PE.

Implications

When the previous conditions are met, a table walk generated for TBE translations might generate transactions using MPAM information associated with the current translation regime rather than the owning translation regime for TBE.

Workaround

This erratum has no workaround.

2247884
A continuous stream of stores might prevent an older store from becoming visible in finite time

Status

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

Description

A continuous stream of stores might prevent an older store from becoming visible in finite time.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs when all the following conditions are met:

- The core executes a store to cacheable address A.
- The core executes a continuous stream of stores at one or more different cacheable addresses.

Implications

If the conditions are met, a store operation executed on the PE where the stream is ongoing might not be visible in finite time.
The core itself keeps performing forward progress and stays interruptible.

Workaround

A denial of service on a Secure process can be avoided if the Secure OS sets up a timer-based interrupt source to interrupt all cores periodically.

2246816
Disabling MPMM through Utility Bus can cause a deadlock

Status

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

Description

Enabling MPMM, through MSR or Utility Bus, then disabling through Utility Bus can cause a deadlock.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under the following conditions:

- The two writes to enable and disable MPMM through CPUMPMMCR_EL3.MPMM_EN must occur in less than 128 cycles.
- Dispatch Throttling (CPUMPMMTUNE_EL3.DT_THR) is set to a value different than 0.

Implications

If the above conditions are met and under specific timing conditions, the PE will deadlock.

Workaround

MPMM is intended to be changed at low frequency (several milliseconds), meaning that it is not expected to be enabled, then disabled in less than 128 cycles under normal operating conditions. In this context, Arm does not expect a workaround to be required.

In the unlikely case that a workaround is required, Dispatch Throttling should be disabled by setting CPUMPMMTUNE_EL3.DT_THR to 0 through the Utility Bus, at least 128 cycles before disabling MPMM.

2238988
A continuous stream of stores might prevent forward progress of a coherency operation

Status

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

Description

A continuous stream of stores targeting the same cache line on a core might prevent forward progress of a coherency operation, leading to a denial of service and perceived deadlock on a different PE or component in the system.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

- The core started a coherency operation in order to perform a store to cacheable address A.
- The core executes a continuous stream of stores at a different cacheable address B.
- The core performs a continuous stream of other L1 cache accesses, preventing stores at address B to perform the write. This might be a series of cache maintenance, loads, or coherency operations received from other PEs.
- The core receives a coherency operation at address A.

Implications

If the conditions are met, the coherency operation might not make forward progress while the store stream is ongoing.
The core itself keeps performing forward progress and stays interruptible.

Workaround

A denial of service on a Secure process can be avoided if the Secure OS sets up a timer-based interrupt source to interrupt all cores periodically.

2238597
A MTE checked store might report a speculative SError in case of poisoned tags

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0 , r1p1 and r1p2. Open.

Description

A Tag-Checked store instruction that observes poison during the tag check operation might report an SError even if the store is not architecturally executed.

Configurations Affected

All configurations with CORE_CACHE_PROTECTION set to True are affected.

Conditions

The erratum occurs under the following conditions:

- The core executes a Tag-Checked store instruction.
- The store observes poison on the allocation tags while performing the tag check operation.
- The store is not architecturally executed, and is discarded as a result of an older branch misprediction, interrupt, or other micro-architectural conditions causing a pipeline flush.

Implications

If the conditions are met, the core might report an SError for an instruction that is not architecturally executed.

This makes the CPU contradict the value of the ID_AA64MMFR1_EL1.SpecSEI, behaving as if it were 0b0001.

Workaround

This erratum has no workaround.

2238671
SVE PRFM instructions fail to prefetch at the expected address

Status

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

Description

Scalable Vector Extension (SVE) PRFM instructions fail to prefetch at the expected address.

Configurations Affected

All configurations are affected.

Conditions

No specific conditions required.

Implications

The previously mentioned operations fail to shift the register argument by 1/2/4. The VA used for prefetching is the sum of the base register and unshifted offset register.

Workaround

There is no workaround.

2227533
APB accesses to unallocated addresses might generate an PSLVERR response

Status

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

Description

An APB access to an unallocated address in the CPU memory map should consider the access RAZ/WI.

Due to this erratum, the CPU correctly treats writes as WI but might respond to the access with a PSLVERR response

Configurations Affected

This erratum affects all the configurations.

Conditions

This erratum occurs when receiving an APB transaction from the external debug to an unallocated CPU memory map address.

Implications

If the above condition is met, the PE will respond to the access with a PLSVERR response. If the APB access was triggered by the PE itself, this will cause the PE to raise an SError Exception.

Workaround

This erratum can be avoided if software avoids accessing addresses that are unallocated in the PE memory map.

2220769
Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work

Status

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

Description

Valid access to ICV_BPR1_EL1 from Secure EL1 when ICV_CTLR_EL1.CBPR is set to 1 should modify ICV_BPR0_EL1 on writes and return the value from ICV_BPR0_EL1 on reads. Instead, reads of ICV_BPR1_EL1 return ICV_BPR0_EL1 plus one, saturated to 0b111. Writes to ICV_BPR1_EL1 are ignored.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The PE is in Secure state in the EL1 exception level.
2. ICV_CTLR_EL1.CBPR is set to 1.
3. A valid read or write access to ICV_BPR1_EL1 occurs.

Implications

If the above conditions are met, then an incorrect value might be returned on read or a valid write might be ignored potentially, affecting the priority of interrupts in the CPU.

Workaround

This erratum has no workaround.

2189618
ELR_ELx might contain an incorrect value on exceptions taken from PC 0xFFFF_0000_0000_0000

Status

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

Description

Upon taking an exception from PC 0xFFFF_0000_0000_0000, the expected ELR_ELx value would be 0xFFFF_0000_0000_0000 or 0xFFFF_0000_0000_0004 depending on the exception type.
When the erratum happens, the ELR_ELx is instead updated with the value 0x0001_0000_0000_0000 or 0x0001_0000_0000_0004 depending on the exception type.

Configurations Affected

All configurations are affected.

Conditions

- The PE is using the EL0&1 or EL0&2 translation regime.
- An exception is taken from PC 0xFFFF_0000_0000_0000.
- Execution at this PC is the result of either:
  - an ESB instruction that synchronizes a pending SError
  - an architectural exception taken at this address, and 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 either:
    - a load or store instruction
    - the instruction opcode at 0xffff_0000_0000_0000 having changed since the last execution at the same PC

Implications

If the above conditions are met, ELR_ELx will be set to 0x0001_0000_0000_0000 or 0x0001_0000_0000_0004 (depending on the exception type) rather than the expected value of 0xFFFF_0000_0000_0000 or 0xFFFF_0000_0000_0004.
Executing an ERET instruction with ELR_ELx set to this incorrect value will trigger an Instruction Abort due to the top bits of the VA being non-canonical.

Workaround

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

If the associated ESR_ELx or exception vector indicates a fault type other than Instruction Abort with a Level 0 Translation Fault/Address Size Fault, the software might assume that the PE hit the erratum and sign-extend ELR_ELx[48] to all the top bits before performing an ERET.

2187565
A double-bit ECC error on the L2 TQ RAM for a store might fail to mark the data as poisoned

Status

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

Description

On a double-bit error resulting from reading the L2 TQ RAM, the corresponding write data is normally marked as poisoned so that a subsequent read to the same location can detect the corruption and report it appropriately.
When the erratum occurs, the data is discarded rather than being marked as poisoned.

Configurations Affected

All configurations are affected.

Conditions

- The affected store uses one of the following memory types:
  - Cacheable Write-Through
  - Cacheable Non-Shared
  - Non-Cacheable
  - Device-GRE
- The double-bit error affects the portion of the TQ RAM containing the byte enable signals, bits <X-Y>.

Implications

If the previous conditions are met, the affected store data will be discarded.
The error will be reported through the RAS reporting logic. Poison will not be signalled, and no SError Exception will be reported.
A subsequent load to the same memory location will return a stale, non-poisoned value from memory.

Workaround

There is no workaround.

2182762
Direct RAM Access read on IData RAM might return corrupted data

Status

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

Description

When the PE is fetching code in normal execution, some linefill in the Level 1 I$ are occurring. These linefills result in allocation of code (that is, write access) into the Idata RAM. Simultaneously, if a Direct RAM Access Read to the IData RAM is executed, a read access is requested to the IData RAM. In some specific conditions, a linefill allocation might occur exactly at the same cycle as a Direct RAM access read is requested. In this case, the PE always grants access to the linefill allocation: the write is done to the IData RAM, and the IData RAM output is thus unknown. However in this erratum, the PE updates nevertheless, and unexpectedly the Direct RAM Access IMP_ISIDE_DATA*_EL3 register with this unknown value, which results in corrupted data being stored.

Configurations Affected

This erratum affects all configurations.

Conditions

A linefill return occurs in the same cycle that the Direct RAM Access Read is arbitrated.

Implications

An MRS <Xt>, IMP_ISIDE_DATA*_EL3 will return a value that does not reflect the content of the targetted RAM.

Workaround

This erratum has no workaround

2180579
Load/Store instructions might fail to report double-bit ECC errors, and/or External aborts, and/or cause loss of poison information

Status

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

Description

When poisoned data is detected in the caches or an External abort received, the error should be reported either through an SError interrupt, or an External abort.
When this erratum happens, the appropriate abort might fail to be reported.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum can occur if one of the following access is performed:

- Store instructions
- Load instructions
- LDG(M) instructions
- Atomic load instructions executed outside of the PE

Implications

- For store instructions, poison information might get permanently cleared from the RAM for the corresponding 64-bytes aligned granule.
- For all other cases listed in previous conditions, poison information, double-bit ECC errors on the data cache, or External abort information from the system might fail to be reported. In such a case, incorrect data are returned and consumed by the PE. Double-bit ECC errors are still correctly logged in the RAS registers. Other errors are silently ignored.

Workaround

This erratum has no workaround.

2171772
A double-bit ECC error or poison on tags on a First-fault load might result in a Synchronous Tag Check Fault instead of a Synchronous External abort

Status

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

Description

When a double-bit error or poison on the tag is encountered, the resulting fault should take precedence over all other faults that might result from the incorrect data being consumed.
When this erratum occurs, Tag Check Fault that is a consequence of the double-bit error or poison might incorrectly take precedence over the expected Synchronous External abort.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs when the following conditions are met on the affected load:

- The First-fault load is a contiguous load.
- The load crosses a cacheline boundary.
- The load is tag-checked and SCTLR_ELx.TCF == 0b01 or 0b11 (Tag Check Faults cause a synchronous exception on loads).
- The load receives a double-bit tag error or poison on both cachelines it accesses.
- The double-bit error or poison causes a corruption of the MTE allocation tag.

Implications

If the above conditions are met, a Data Abort exception with DFSC == 0b010001 (Synchronous Tag Check Fault) might be reported instead of DFSC == 0b010000 (Synchronous External abort, not on translation table walk or hardware update of translation table).
The tag error is still correctly detected and logged in the RAS error reporting registers.

Workaround

There is no workaround.

2168152
Persistent errors on SVE First-fault instructions might result in deadlock under specific conditions

Status

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

Description

Persistent errors on Scalable Vector Extension (SVE) First-fault load instructions might result in deadlock when the instruction triggers other specific fault conditions.

Configurations Affected

All configurations are affected.

Conditions

- The affected load is a Memory Tagging Extension (MTE) tag-checked load.
- The affected load triggers a watchpoint on an element other than the first active element.
- The affected load always receives an error response when bringing the line into the L1 cache. This can be caused by either:
  - an error response from the system
  - a permanent error in the L2 cache

Implications

If the previous conditions are met, the core will stop processing new instructions and interrupts and might stop servicing incoming snoop requests, leading to deadlock.

Workaround

There is no workaround.

2160316
Trace data might miss some packets or contain incorrect packets when enabled through the APB interface

Status

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

Description

When tracing is enabled by System register writes done through the APB memory-mapped interface, the generated trace might miss packets or output incorrect packets for a limited period of time.

Configurations Affected

All configurations are affected.

Conditions

This erratum might occur when trace is enabled by writes through the memory-mapped interface.

Software that enables trace using MSR instructions is not affected by this erratum.

Implications

The output trace might miss some packets or contain incorrect packets after it has been enabled. Up to 512 branch packets might be incorrect following the first output branch packet. After these initial 512 packets, the trace data will be correct and consistent with the actual PE execution.

Trace unit resources such as comparators might fire incorrectly.

No prohibited information is output.

Workaround

The erratum does not occur if the trace unit is enabled while the PE is in Debug state.