Arm Neoverse V1 (MP076)

Software Developer Errata Notice

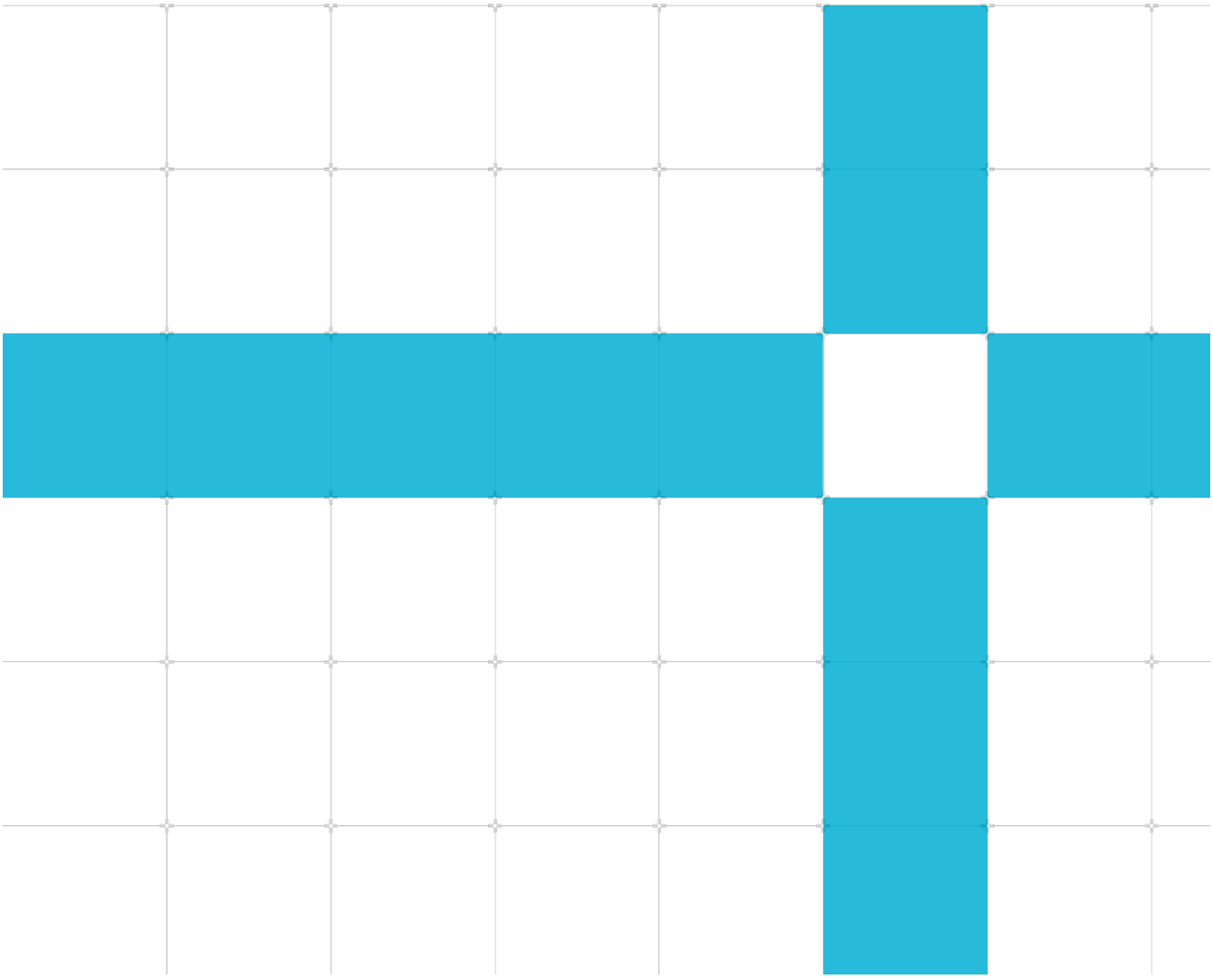
Date of issue: October 01, 2024

Non-Confidential

Document version: 20.0

Document ID: SDEN-1401781

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



This document (SDEN_1401781_20.0_en) was issued on October 01, 2024.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-1401781**

Inclusive language commitment

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

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Neoverse V1 (MP076), create a ticket on **https://support.developer.arm.com**.

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

Contents

r0p0 implementation fixes

Note the following errata might be fixed in some implementations of r0p0. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR_EL1[0]	1618629 Vector instructions might cause deadlock under specific micro-architectural conditions

Note that there is no change to the MIDR_EL1 which remains at r0p0 but the REVIDR_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

r1p1 implementation fixes

Note the following errata might be fixed in some implementations of r1p1. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR_EL1[0]	1925756 Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state

Note that there is no change to the MIDR_EL1 which remains at r1p1, but the REVIDR_EL1 is updated from 0x0 to 0x1 to indicate that the new erratum is fixed (via REVIDR_EL1[0]). Software will identify this release through a 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.

October 01, 2024: Changes in document version v20.0

ID	Status	Area	Category	Summary
3696285	New	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock
3605044	New	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed
3607341	New	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative
3627242	New	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability
3633452	New	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception
3640938	New	Programmer	Category C	SPE operation type is corrupted under certain conditions
3694440	New	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled
3694464	New	Programmer	Category C	FFR might not capture the lowest faulting memory element
3700174	New	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors
3705910	New	Programmer	Category C	PMU events are mis-categorized by not considering the effect of "Taken locally"

April 30, 2024: Changes in document version v19.0

ID	Status	Area	Category	Summary
3324341	New	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing

August 18, 2023: Changes in document version v18.0

ID	Status	Area	Category	Summary
3003013	New	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly
3028884	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
2986644	New	Programmer	Category B (rare)	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level
2795125	New	Programmer	Category C	DGH instruction doesn't execute correctly
2910962	New	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts
2985982	New	Programmer	Category C	SPE latency counters are corrupted under certain conditions

February 22, 2023: Changes in document version v17.0

ID	Status	Area	Category	Summary
2798806	New	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions
2816902	New	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM

November 09, 2022: Changes in document version v16.0

ID	Status	Area	Category	Summary
2743233	New	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock
2743093	New	Programmer	Category B	The core might deadlock during powerdown sequence
2779461	New	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation
2755353	New	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

July 27, 2022: Changes in document version v15.0

ID	Status	Area	Category	Summary
2675258	New	Programmer	Category B	Static and dynamic TXREQ limiting might cause deadlock
2701953	New	Programmer	Category B	Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back
2348377	Updated	Programmer	Category B (rare)	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment
1858170	New	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED
2183513	New	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison
2238112	New	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State
2239141	New	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state
2278133	New	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate
2444422	New	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly
2647275	New	Programmer	Category C	Incorrect read value for Performance Monitors Control Register
2707725	New	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field

January 21, 2022: Changes in document version v14.0

ID	Status	Area	Category	Summary
2294912	New	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
2372203	New	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption
2348377	New	Programmer	Category B (rare)	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment
2276445	New	Programmer	Category C	SPE, PMU event for full/partial/empty/not full predicate might be incorrect for some SVE instructions
2285233	New	Programmer	Category C	Lower priority exception might be reported when abort condition is detected at both stages of translation
2307835	New	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk
2391681	New	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR

September 24, 2021: Changes in document version v13.0

ID	Status	Area	Category	Summary
1966096	Updated	Programmer	Category B	The instruction following a change in SVE Vector length in ZCR_EL1 might not see updated value
2139242	Updated	Programmer	Category B	Hardware data prefetcher can generate a physical address with incorrect attributes
2216392	New	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion
2261535	New	Programmer	Category C	L1 Data poison is not cleared by a store
2262865	New	Programmer	Category C	Incorrect sampling of SPE event "Partial predicate" for SVE instruction with no vector operands

May 27, 2021: Changes in document version v12.0

ID	Status	Area	Category	Summary
1863563	Updated	Programmer	Category B	Core might generate breakpoint exception on incorrect IA
2108267	New	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock
2139242	New	Programmer	Category B	Hardware data prefetcher can generate a physical address with incorrect attributes
1794808	Updated	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock
1986030	New	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes
2089669	New	Programmer	Category C	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain
2134909	New	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect
2137418	New	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely
2143135	New	Programmer	Category C	Some SVE PMU events count incorrectly

March 03, 2021: Changes in document version v11.0

ID	Status	Area	Category	Summary
2020786	New	Programmer	Category C	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation
2033528	New	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom
2052426	New	Programmer	Category C	An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers
2070947	New	Programmer	Category C	External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register

December 09, 2020: Changes in document version v10.0

ID	Status	Area	Category	Summary
1863563	Updated	Programmer	Category B	Core might generate breakpoint exception on incorrect IA
1949546	Updated	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State

November 13, 2020: Changes in document version v9.0

ID	Status	Area	Category	Summary
1915893	New	Programmer	Category B	Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled
1978082	New	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock
1937142	New	Programmer	Category C	MPAM value associated with translation table walk request might be incorrect
1960366	New	Programmer	Category C	Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC
1965494	New	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled
1987045	New	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFI or WFE state
2001721	New	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL

September 24, 2020: Changes in document version v8.0

ID	Status	Area	Category	Summary
1618635	Updated	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
1923200	New	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset
1925756	New	Programmer	Category B	Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state
1940577	New	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics
1966096	New	Programmer	Category B	The instruction following a change in SVE Vector length in ZCR_EL1 might not see updated value
1853757	New	Programmer	Category C	The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified
1872062	New	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors
1878813	New	Programmer	Category C	PFG duplicate reported faults through a Warm reset
1880117	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events
1890031	New	Programmer	Category C	Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO
1923196	New	Programmer	Category C	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB
1925786	New	Programmer	Category C	Unsupported atomic fault due to memory type defined in first stage of translation might result in exception being taken to EL2
1949546	New	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State

June 17, 2020: Changes in document version v7.0

ID	Status	Area	Category	Summary
1659792	Updated	Programmer	Category B	Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation
1659794	Updated	Programmer	Category B	Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer
1774420	New	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB
1791573	New	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures
1852267	New	Programmer	Category B	SPE might illegally write to any physical address, including Secure space
1863563	New	Programmer	Category B	Core might generate breakpoint exception on incorrect IA
1690750	Updated	Programmer	Category C	The core might record incorrect INDEX into ERR0MISC0 when L0 Macro-op cache is affected by parity error

ID	Status	Area	Category	Summary
1694300	Updated	Programmer	Category C	Instruction sampling bias exists in SPE implementation
1702491	Updated	Programmer	Category C	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB
1702951	Updated	Programmer	Category C	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice
1740839	Updated	Programmer	Category C	RAS error reported could have incorrect value in ERR0ADDR_EL1
1755651	New	Programmer	Category C	MPAM value associated with descriptor fetch requests could be incorrect
1771937	New	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR
1776362	New	Programmer	Category C	A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data
1778872	New	Programmer	Category C	Possible loss of CTI event
1784478	New	Programmer	Category C	Loss of CTI events during warm reset
1791399	New	Programmer	Category C	The core might deadlock when an external debugger injects instructions using ITR register
1794808	New	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock
1803672	New	Programmer	Category C	Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents
1817602	New	Programmer	Category C	Persistent faults on speculative elements of SVE First-fault gather-load instructions might result in deadlock
1825527	New	Programmer	Category C	Transient L2 tag double bit Errors might cause data corruption
1825496	New	Programmer	Category C	External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior
1835961	New	Programmer	Category C	MPAM value associated with MMU descriptor fetch requests might be incorrect
1846398	New	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect

February 14, 2020: Changes in document version v6.0

ID	Status	Area	Category	Summary
1659792	New	Programmer	Category B	Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation
1659794	New	Programmer	Category B	Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer
1657962	New	Programmer	Category C	Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock
1690750	New	Programmer	Category C	The core might record incorrect INDEX into ERR0MISC0 when L0 Macro-op cache is affected by parity error
1694300	New	Programmer	Category C	Instruction sampling bias exists in SPE implementation
1702491	New	Programmer	Category C	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB
1702951	New	Programmer	Category C	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice
1740839	New	Programmer	Category C	RAS error reported could have incorrect value in ERR0ADDR_EL1

December 10, 2019: Changes in document version v5.0

ID	Status	Area	Category	Summary
1618629	Updated	Programmer	Category A	Vector instructions might cause deadlock under specific micro-architectural conditions
1625573	Updated	Programmer	Category A	PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk
1445139	Updated	Programmer	Category B	A DC CVAU or IC CVAU instruction behaves like a NOP
1466307	Updated	Programmer	Category B	Self synchronizing behavior of changing effective vector length via the ZCR_ELx might not take effect until after the following instruction, rather than before as architecturally required
1508565	Updated	Programmer	Category B	Accessing a memory location using mismatched shareability attributes might cause loss of coherency
1522191	Updated	Programmer	Category B	Corruption of cumulative floating point exception bits
1542436	Updated	Programmer	Category B	Corruption of scalable vector register following execution of an SVE prefixed Integer Multiply instruction
1592691	New	Programmer	Category B	ICH_HCR_EL2.vSGIEOICOUNT masks end of interrupt counting for some virtual Private Peripheral Interrupts
1618630	Updated	Programmer	Category B	Branch prediction for an ERET cached in the instruction cache might cause a deadlock
1618634	Updated	Programmer	Category B	Incorrect instructions might be executed
1618635	Updated	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock

ID	Status	Area	Category	Summary
1618636	Updated	Programmer	Category B	Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core
1619807	Updated	Programmer	Category B	The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache
1619814	Updated	Programmer	Category B	Enabling SPE might result in deadlock in some situations
1654562	New	Programmer	Category B	A streaming write in the presence of a store-release instruction might result in data corruption
1674403	New	Programmer	Category B	The core might deadlock when executing a program sequence with divide or square-root operations
1543964	Updated	Programmer	Category B (rare)	The core might fetch a stale instruction from memory which violates the ordering of instruction fetches
1618631	Updated	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
1452392	Updated	Programmer	Category C	Unaligned SVE Non-Fault load that encounters a tag double-bit error or data poison might resolve with an abort
1507873	Updated	Programmer	Category C	Misaligned atomics to unsupported Non-cacheable or Device memory location might lead to incorrect DFSC code in the ESR
1549423	Updated	Programmer	Category C	Transient parity error in L1 instruction cache might result in missed breakpoint exception
1553581	Updated	Programmer	Category C	ID_PFR1_EL1.GIC incorrectly reports as 0x1
1591505	Updated	Programmer	Category C	Predicate event information populated incorrectly for non-predicated SVE instructions sampled by SPE
1595153	New	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect
1596291	New	Programmer	Category C	Data side MPAM values used for PLI
1618632	Updated	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
1618633	Updated	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled
1618637	Updated	Programmer	Category C	Interrupt might be taken later than architecturally mandated on exit from Debug state
1619800	Updated	Programmer	Category C	An unaligned load may initiate a prefetch request which crosses a page boundary
1619801	Updated	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect
1619805	Updated	Programmer	Category C	ESB instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE
1619808	Updated	Programmer	Category C	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error
1619809	Updated	Programmer	Category C	PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled

ID	Status	Area	Category	Summary
1619811	Updated	Programmer	Category C	The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception
1619813	Updated	Programmer	Category C	Enabling L2 cache partitioning might result in a loss of performance
1619815	Updated	Programmer	Category C	ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort
1619816	Updated	Programmer	Category C	A load to normal memory might trigger a prefetch request outside of the current mapped page
1619817	Updated	Programmer	Category C	RAS error status records could log spurious corrected error
1629078	New	Programmer	Category C	ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly
1651942	New	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption
1656369	New	Programmer	Category C	APB access to trace registers does not work during Warm reset
1659168	New	Programmer	Category C	A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERR0MISC0_EL1 and ER0ADDR_EL1
1662728	New	Programmer	Category C	Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior

October 14, 2019: Changes in document version v4.0

ID	Status	Area	Category	Summary
1625573	New	Programmer	Category A	PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk
1508565	New	Programmer	Category B	Accessing a memory location using mismatched shareability attributes might cause loss of coherency
1522191	New	Programmer	Category B	Corruption of cumulative floating point exception bits
1542436	New	Programmer	Category B	Corruption of scalable vector register following execution of an SVE prefixed Integer Multiply instruction
1619814	New	Programmer	Category B	Enabling SPE might result in deadlock in some situations
1543964	New	Programmer	Category B (rare)	The core might fetch a stale instruction from memory which violates the ordering of instruction fetches
1553581	New	Programmer	Category C	ID_PFR1_EL1.GIC incorrectly reports as 0x1
1591505	New	Programmer	Category C	Predicate event information populated incorrectly for non-predicated SVE instructions sampled by SPE
1619805	New	Programmer	Category C	ESB instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE
1619808	New	Programmer	Category C	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error
1619809	New	Programmer	Category C	PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled
1619811	New	Programmer	Category C	The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception
1619813	New	Programmer	Category C	Enabling L2 cache partitioning might result in a loss of performance
1619815	New	Programmer	Category C	ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort
1619816	New	Programmer	Category C	A load to normal memory might trigger a prefetch request outside of the current mapped page
1619817	New	Programmer	Category C	RAS error status records could log spurious corrected error

July 19, 2019: Changes in document version v3.0

ID	Status	Area	Category	Summary
1466307	New	Programmer	Category B	Self synchronizing behavior of changing effective vector length via the ZCR_ELx might not take effect until after the following instruction, rather than before as architecturally required
1618634	New	Programmer	Category B	Incorrect instructions might be executed
1618635	New	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
1618636	New	Programmer	Category B	Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core
1619807	New	Programmer	Category B	The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache
1507873	New	Programmer	Category C	Misaligned atomics to unsupported Non-cacheable or Device memory location might lead to incorrect DFSC code in the ESR
1549423	New	Programmer	Category C	Transient parity error in L1 instruction cache might result in missed breakpoint exception
1618637	New	Programmer	Category C	Interrupt might be taken later than architecturally mandated on exit from Debug state
1619800	New	Programmer	Category C	An unaligned load may initiate a prefetch request which crosses a page boundary
1619801	New	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect

May 17, 2019: Changes in document version v2.0

ID	Status	Area	Category	Summary
1618629	New	Programmer	Category A	Vector instructions might cause deadlock under specific micro-architectural conditions
1445139	New	Programmer	Category B	A DC CVAU or IC CVAU instruction behaves like a NOP
1618630	New	Programmer	Category B	Branch prediction for an ERET cached in the instruction cache might cause a deadlock
1618631	New	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
1452392	New	Programmer	Category C	Unaligned SVE Non-Fault load that encounters a tag double-bit error or data poison might resolve with an abort
1618632	New	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
1618633	New	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled

March 27, 2019: 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
1618629	Programmer	Category A	Vector instructions might cause deadlock under specific micro-architectural conditions	r0p0	r1p0
1625573	Programmer	Category A	PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk	r0p0	r1p0
1445139	Programmer	Category B	A DC CVAU or IC CVAU instruction behaves like a NOP	r0p0	r1p0
1466307	Programmer	Category B	Self synchronizing behavior of changing effective vector length via the ZCR_ELx might not take effect until after the following instruction, rather than before as architecturally required	r0p0	r1p0
1508565	Programmer	Category B	Accessing a memory location using mismatched shareability attributes might cause loss of coherency	r0p0	r1p0
1522191	Programmer	Category B	Corruption of cumulative floating point exception bits	r0p0	r1p0
1542436	Programmer	Category B	Corruption of scalable vector register following execution of an SVE prefixed Integer Multiply instruction	r0p0	r1p0
1592691	Programmer	Category B	ICH_HCR_EL2.vSGIEOICOUNT masks end of interrupt counting for some virtual Private Peripheral Interrupts	r0p0	r1p0
1618630	Programmer	Category B	Branch prediction for an ERET cached in the instruction cache might cause a deadlock	r0p0	r1p0
1618634	Programmer	Category B	Incorrect instructions might be executed	r0p0	r1p0
1618635	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock	r0p0	r1p0
1618636	Programmer	Category B	Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
1619807	Programmer	Category B	The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache	r0p0	r1p0
1619814	Programmer	Category B	Enabling SPE might result in deadlock in some situations	r0p0	r1p0
1654562	Programmer	Category B	A streaming write in the presence of a store-release instruction might result in data corruption	r0p0	r1p0
1659792	Programmer	Category B	Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation	r0p0, r1p0	r1p1
1659794	Programmer	Category B	Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer	r0p0, r1p0	r1p1
1674403	Programmer	Category B	The core might deadlock when executing a program sequence with divide or square-root operations	r0p0	r1p0
1774420	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB	r0p0, r1p0	r1p1
1791573	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures	r0p0, r1p0	r1p1
1852267	Programmer	Category B	SPE might illegally write to any physical address, including Secure space	r0p0, r1p0	r1p1
1863563	Programmer	Category B	Core might generate breakpoint exception on incorrect IA	r0p0, r1p0	r1p1
1915893	Programmer	Category B	Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled	r0p0, r1p0, r1p1	r1p2
1923200	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset	r0p0, r1p0, r1p1	r1p2

ID	Area	Category	Summary	Found in versions	Fixed in version
1925756	Programmer	Category B	Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state	r0p0, r1p0, r1p1	r1p2
1940577	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics	r0p0, r1p0, r1p1	r1p2
1966096	Programmer	Category B	The instruction following a change in SVE Vector length in ZCR_EL1 might not see updated value	r0p0, r1p0, r1p1	r1p2
1978082	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock	r0p0, r1p0, r1p1	r1p2
2108267	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock	r0p0, r1p0, r1p1, r1p2	Open
2139242	Programmer	Category B	Hardware data prefetcher can generate a physical address with incorrect attributes	r0p0, r1p0, r1p1	r1p2
2216392	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion	r0p0, r1p0, r1p1	r1p2
2294912	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	r0p0, r1p0, r1p1, r1p2	Open
2372203	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption	r0p0, r1p0, r1p1	r1p2
2675258	Programmer	Category B	Static and dynamic TXREQ limiting might cause deadlock	r0p0, r1p0, r1p1	r1p2
2701953	Programmer	Category B	Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back	r0p0, r1p0, r1p1	r1p2
2743093	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r1p0, r1p1, r1p2	Open
2743233	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock	r0p0, r1p0, r1p1, r1p2	Open
2779461	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation	r0p0, r1p0, r1p1, r1p2	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3003013	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly	r0p0, r1p0, r1p1, r1p2	Open
3028884	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0, r1p0, r1p1, r1p2	Open
3324341	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	r0p0, r1p0, r1p1, r1p2	Open
3696285	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	r0p0, r1p0, r1p1, r1p2	Open
1543964	Programmer	Category B (rare)	The core might fetch a stale instruction from memory which violates the ordering of instruction fetches	r0p0	r1p0
1618631	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data	r0p0	r1p0
2348377	Programmer	Category B (rare)	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment	r0p0, r1p0, r1p1	r1p2
2986644	Programmer	Category B (rare)	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level	r0p0, r1p0, r1p1, r1p2	Open
1452392	Programmer	Category C	Unaligned SVE Non-Fault load that encounters a tag double-bit error or data poison might resolve with an abort	r0p0	r1p0
1507873	Programmer	Category C	Misaligned atomics to unsupported Non-cacheable or Device memory location might lead to incorrect DFSC code in the ESR	r0p0	r1p0
1549423	Programmer	Category C	Transient parity error in L1 instruction cache might result in missed breakpoint exception	r0p0	r1p0
1553581	Programmer	Category C	ID_PFR1_EL1.GIC incorrectly reports as 0x1	r0p0	r1p0
1591505	Programmer	Category C	Predicate event information populated incorrectly for non-predicated SVE instructions sampled by SPE	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
1595153	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect	r0p0	r1p0
1596291	Programmer	Category C	Data side MPAM values used for PLI	r0p0	r1p0
1618632	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0	r0p0	r1p0
1618633	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled	r0p0	r1p0
1618637	Programmer	Category C	Interrupt might be taken later than architecturally mandated on exit from Debug state	r0p0	r1p0
1619800	Programmer	Category C	An unaligned load may initiate a prefetch request which crosses a page boundary	r0p0	r1p0
1619801	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect	r0p0	r1p0
1619805	Programmer	Category C	ESB instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE	r0p0	r1p0
1619808	Programmer	Category C	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error	r0p0	r1p0
1619809	Programmer	Category C	PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled	r0p0	r1p0
1619811	Programmer	Category C	The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception	r0p0	r1p0
1619813	Programmer	Category C	Enabling L2 cache partitioning might result in a loss of performance	r0p0	r1p0
1619815	Programmer	Category C	ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort	r0p0	r1p0
1619816	Programmer	Category C	A load to normal memory might trigger a prefetch request outside of the current mapped page	r0p0	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
1619817	Programmer	Category C	RAS error status records could log spurious corrected error	r0p0	r1p0
1629078	Programmer	Category C	ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly	r0p0	r1p0
1651942	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption	r0p0	r1p0
1656369	Programmer	Category C	APB access to trace registers does not work during Warm reset	r0p0	r1p0
1657962	Programmer	Category C	Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock	r0p0	r1p0
1659168	Programmer	Category C	A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERR0MISC0_EL1 and ER0ADDR_EL1	r0p0	r1p0
1662728	Programmer	Category C	Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior	r0p0	r1p0
1690750	Programmer	Category C	The core might record incorrect INDEX into ERR0MISC0 when L0 Macro-op cache is affected by parity error	r0p0, r1p0	r1p1
1694300	Programmer	Category C	Instruction sampling bias exists in SPE implementation	r0p0, r1p0	r1p1
1702491	Programmer	Category C	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB	r0p0, r1p0	r1p1
1702951	Programmer	Category C	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice	r0p0, r1p0	r1p1
1740839	Programmer	Category C	RAS error reported could have incorrect value in ERR0ADDR_EL1	r0p0, r1p0	r1p1
1755651	Programmer	Category C	MPAM value associated with descriptor fetch requests could be incorrect	r0p0, r1p0	r1p1
1771937	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR	r0p0, r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
1776362	Programmer	Category C	A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data	r0p0, r1p0	r1p1
1778872	Programmer	Category C	Possible loss of CTI event	r0p0, r1p0	r1p1
1784478	Programmer	Category C	Loss of CTI events during warm reset	r0p0, r1p0	r1p1
1791399	Programmer	Category C	The core might deadlock when an external debugger injects instructions using ITR register	r0p0, r1p0	r1p1
1794808	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock	r0p0, r1p0	r1p1
1803672	Programmer	Category C	Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents	r0p0, r1p0	r1p1
1817602	Programmer	Category C	Persistent faults on speculative elements of SVE First-fault gather-load instructions might result in deadlock	r0p0, r1p0	r1p1
1825496	Programmer	Category C	External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior	r0p0, r1p0	r1p1
1825527	Programmer	Category C	Transient L2 tag double bit Errors might cause data corruption	r0p0, r1p0	r1p1
1835961	Programmer	Category C	MPAM value associated with MMU descriptor fetch requests might be incorrect	r0p0, r1p0	r1p1
1846398	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect	r0p0, r1p0	r1p1
1853757	Programmer	Category C	The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified	r0p0, r1p0, r1p1, r1p2	Open
1858170	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED	r0p0, r1p0, r1p1	r1p2
1872062	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors	r0p0, r1p0, r1p1	r1p2

ID	Area	Category	Summary	Found in versions	Fixed in version
1878813	Programmer	Category C	PFG duplicate reported faults through a Warm reset	r0p0, r1p0, r1p1	r1p2
1880117	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0, r1p0, r1p1, r1p2	Open
1890031	Programmer	Category C	Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO	r0p0, r1p0, r1p1	r1p2
1923196	Programmer	Category C	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB	r0p0, r1p0, r1p1, r1p2	Open
1925786	Programmer	Category C	Unsupported atomic fault due to memory type defined in first stage of translation might result in exception being taken to EL2	r0p0, r1p0, r1p1	r1p2
1937142	Programmer	Category C	MPAM value associated with translation table walk request might be incorrect	r0p0, r1p0, r1p1, r1p2	Open
1949546	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State	r0p0, r1p0, r1p1	r1p2
1960366	Programmer	Category C	Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC	r0p0, r1p0, r1p1	r1p2
1965494	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	r0p0, r1p0, r1p1	r1p2
1986030	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes	r0p0, r1p0, r1p1	r1p2
1987045	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFI or WFE state	r0p0, r1p0, r1p1	r1p2
2001721	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL	r0p0, r1p0, r1p1	r1p2
2020786	Programmer	Category C	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation	r0p0, r1p0, r1p1	r1p2
2033528	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom	r0p0, r1p0, r1p1	r1p2

ID	Area	Category	Summary	Found in versions	Fixed in version
2052426	Programmer	Category C	An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers	r0p0, r1p0, r1p1, r1p2	Open
2070947	Programmer	Category C	External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register	r0p0, r1p0, r1p1	r1p2
2089669	Programmer	Category C	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain	r0p0, r1p0, r1p1	r1p2
2134909	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect	r0p0, r1p0, r1p1, r1p2	Open
2137418	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely	r0p0, r1p0, r1p1, r1p2	Open
2143135	Programmer	Category C	Some SVE PMU events count incorrectly	r0p0, r1p0, r1p1, r1p2	Open
2183513	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison	r0p0, r1p0, r1p1	r1p2
2238112	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State	r0p0, r1p0, r1p1, r1p2	Open
2239141	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state	r0p0, r1p0, r1p1, r1p2	Open
2261535	Programmer	Category C	L1 Data poison is not cleared by a store	r0p0, r1p0, r1p1	r1p2
2262865	Programmer	Category C	Incorrect sampling of SPE event "Partial predicate" for SVE instruction with no vector operands	r0p0, r1p0, r1p1	r1p2
2276445	Programmer	Category C	SPE, PMU event for full/partial/empty/not full predicate might be incorrect for some SVE instructions	r0p0, r1p0, r1p1	r1p2
2278133	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate	r0p0, r1p0, r1p1	r1p2
2285233	Programmer	Category C	Lower priority exception might be reported when abort condition is detected at both stages of translation	r0p0, r1p0, r1p1	r1p2
2307835	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk	r0p0, r1p0, r1p1	r1p2

ID	Area	Category	Summary	Found in versions	Fixed in version
2391681	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR	r0p0, r1p0, r1p1, r1p2	Open
2444422	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly	r0p0, r1p0, r1p1	r1p2
2647275	Programmer	Category C	Incorrect read value for Performance Monitors Control Register	r0p0, r1p0, r1p1	r1p2
2707725	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field	r0p0, r1p0, r1p1	r1p2
2755353	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	r0p0, r1p0, r1p1, r1p2	Open
2795125	Programmer	Category C	DGH instruction doesn't execute correctly	r0p0, r1p0, r1p1, r1p2	Open
2798806	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions	r0p0, r1p0, r1p1, r1p2	Open
2816902	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	r0p0, r1p0, r1p1, r1p2	Open
2910962	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts	r0p0, r1p0, r1p1, r1p2	Open
2985982	Programmer	Category C	SPE latency counters are corrupted under certain conditions	r0p0, r1p0, r1p1, r1p2	Open
3605044	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed	r0p0, r1p0, r1p1, r1p2	Open
3607341	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative	r0p0, r1p0, r1p1, r1p2	Open
3627242	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability	r0p0, r1p0, r1p1, r1p2	Open
3633452	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception	r0p0, r1p0, r1p1, r1p2	Open
3640938	Programmer	Category C	SPE operation type is corrupted under certain conditions	r0p0, r1p0, r1p1, r1p2	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3694440	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled	r0p0, r1p0, r1p1, r1p2	Open
3694464	Programmer	Category C	FFR might not capture the lowest faulting memory element	r0p0, r1p0, r1p1, r1p2	Open
3700174	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors	r0p0, r1p0, r1p1, r1p2	Open
3705910	Programmer	Category C	PMU events are mis-categorized by not considering the effect of "Taken locally"	r0p0, r1p0, r1p1, r1p2	Open

Errata descriptions

Category A

1618629
Vector instructions might cause deadlock under specific micro-architectural conditions

Status

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

Description

Under specific micro-architectural conditions, code sequences including Vector instructions can result in a deadlock in the register renaming block of the core.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A vector instruction is executed.
2. Specific micro-architectural conditions occur during register renaming.

Implications

If the above conditions are met, then this erratum might result in a deadlock.

Workaround

There is no workaround.

1625573
PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk

Status

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

Description

When the core fetches instructions from mop-cache, the instruction might corrupt the PC value after the instruction is executed.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in AArch64 state.
2. An instruction fetch detects an instruction TLB miss.
3. The instruction fetch hits in the L0 Macro-op cache after the tablewalk request was sent out to MMU.
4. Subsequent instruction fetches hit in L0 Macro-op cache continuously.
5. The core executes and commits all L0 Macro-op cache hit instructions that were fetched in step 3, before the core receives the address translation response for first TLB miss.

Implications

If the above conditions are met, then one of following implications might occur:

1. PC register might contain the instruction address of the instruction that was fetched in step 3 of the above conditions, which might be incorrect. The core might fetch the wrong instruction based on this incorrect address.
2. ELR register might contain the instruction address of the instruction that was fetched in step 3 of the above conditions, which might be incorrect if the core processes an exception entry. The core might load this corrupted address into PC at the subsequent exception return.

Workaround

There is no workaround.

Category A (rare)

There are no errata in this category.

Category B

1445139
A DC CVAU or IC CVAU instruction behaves like a NOP

Status

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

Description

When operating at Non-secure EL0, the HCR_EL2.TOCU trap bit can affect the execution of DC CVAU and IC CVAU instructions improperly. This occurs when HCR_EL2.{E2H,TGE}=(1,1) and HCR_EL2.TOCU=1. Under these conditions, the DC CVAU and IC CVAU instructions behave as NOPs rather than being executed correctly.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The PE is operating at Non-secure EL0 with HCR_EL2.{E2H,TGE}=(1,1) and HCR_EL2.TOCU=1.
2. The PE executes a DC CVAU or IC CVAU instruction.

Implications

If the above conditions are met, then the DC CVAU or IC CVAU instruction does not execute correctly and instead behaves like a NOP.

Workaround

The HCR_EL2.TOCU bit is not intended to have any effect when HCR_EL2.(E2H,TGE)=(1,1). As such this erratum can be avoided by not setting HCR_EL2.TOCU=1 under these conditions.

1466307
Self synchronizing behavior of changing effective vector length via the ZCR_ELx might not take effect until after the following instruction, rather than before as architecturally required

Status

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

Description

When performing a system register access using MSR(write) instructions to ZCR_ELx, that updates the effective vector length, might not force its synchronization event before the next instruction as architecturally required. The next instruction would then have been executed in the previous effective length setting.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Mop cache must be enabled.
2. Program must be running aarch64 code.
3. Program executes MSR(write) to system register to ZCR_ELx, which changes the effective vector length. A change to effective vector length architecturally requires a self synchronization operation.
4. The following instruction is an SVE instruction that would use the vector length setting.

Implications

The instruction following the self synchronizing MSR(write) of ZCR_ELx could be executed in previous context with incorrect vector length resulting in incorrect operation.

Workaround

Please contact Arm support if implementing the following in r0p0.

This erratum can be avoided through the following write sequence to several IMPLEMENTATION DEFINED registers:

```
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,= 0xEE010F12
```

```
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,= 0xFF1F0FFF
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x80000000003FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
ISB
```

1508565
Accessing a memory location using mismatched shareability attributes might cause loss of coherency

Status

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

Description

If two PEs access the same memory location with mismatched shareability attributes, accesses performed by a PE using a shareable mapping might cause data corruption in a PE using a non-shareable mapping of the same physical address. Similarly, stashing into a PE whose caches were allocated using a non-shareable mapping might cause data corruption.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. PE0 accesses a memory location using cacheable and non-shareable attributes and later writes to the same location, resulting in dirty data in the PE's caches.
2. PE1 accesses the same memory location using cacheable and shareable attributes. Accesses include reads, writes, and cache maintenance operations.
3. Interconnect does not filter snoop traffic to PEs and as a result snoops PE0.
4. Snoop to PE0 causes a cache state change but does not cause a copyback of PE0's dirty data or a stash snoop to PE0 that might request data using a DataPull response, but internal queues are not enabled to expect data.

Implications

If the above conditions are met, PE0 might experience data corruption.

Workaround

Avoid using mismatched shareability attributes for aliases of the same memory location.

1522191
Corruption of cumulative floating point exception bits

Status

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

Description

Under certain circumstances, floating point and Advanced SIMD instructions might record cumulative floating exception bits in the FPSR (AArch64) or FPSCR (AArch32) in a manner that violates the simple sequential execution model.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. Execution of a floating point or Advanced SIMD instruction that indirectly sets a cumulative floating point exception bit (such as IOC, DZC, OFC, UFC, IXC, IDC, or QC).
2. Execution of a floating point status register direct write or direct read instruction, for example MSR/MRS FPSR (AArch64) or VMSR/VMRS FPSCR (AArch32), occurs in close proximity. Note that this is with the exception of "VMRS APSR_nzcv, FPSCR" in AArch32 execution state which is not affected by this erratum.

Implications

If the above conditions are met, then under specific microarchitectural timing conditions the indirect setting of the cumulative floating point exception bit by execution of a floating point instruction might occur out of order with respect to the direct write (MSR) or direct read (MRS) of cumulative floating point exception bits. This leads to the corruption of the architected state of the floating point exception bits.

Workaround

To avoid this erratum, serialize before all direct reads and writes to the FPSR (AArch64) and FPSCR (AArch32), with the exception of "VMRS APSR_nzcv, FPSCR" in AArch32 execution state (not affected by this erratum). This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers accessible only at EL3:

```
LDR x0,=0x5
MSR S3_6_c15_c8_0,x0
```

```
LDR x0,=0xEEE10A10
MSR S3_6_c15_c8_2,x0
LDR x0,=0xFFEF0FFF
MSR S3_6_c15_c8_3,x0
LDR x0,=0x0010F000
MSR S3_6_c15_c8_4,x0
LDR x0,=0x0010F000
MSR S3_6_c15_c8_5,x0
LDR x0,=0x40000080023ff
MSR S3_6_c15_c8_1,x0

LDR x0,=0x6
MSR S3_6_c15_c8_0,x0
LDR x0,=0xEE640F34
MSR S3_6_c15_c8_2,x0
LDR x0,=0xFFEF0FFF
MSR S3_6_c15_c8_3,x0
LDR x0,=0x40000080023ff
MSR S3_6_c15_c8_1,x0

ISB
```

1542436
Corruption of scalable vector register following execution of an SVE prefixed Integer Multiply instruction

Status

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

Description

Under certain circumstances, an SVE prefixed Integer Multiply instruction might return an incorrect result and corrupt its scalable vector destination register.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. An SVE UMULH or SMULH instruction is executed.
2. An SVE MOVPRFX instruction is executed to prefix a younger SVE Integer Multiply instruction in close proximity.
3. The destination scalable vector register of the prefixed instruction is the same as the initial UMULH or SMULH instruction.

Implications

If the above conditions are met, then under specific microarchitectural timing conditions a computation error might occur, thus corrupting the architectural state of the destination scalable vector register.

Workaround

A workaround exists that could have a slight material impact on SVE prefixing with MOVPRFX performance. Boot code must include write to set CPUACTLR4_EL1[14].

1592691 ICH_HCR_EL2.vSGIEOICOUNT masks end of interrupt counting for some virtual Private Peripheral Interrupts

Status

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

Description

The GICv4.1 architecture adds a feature to mask ICH_HCR_EL2.EOIcount from incrementing due to the deactivation of virtual Software Generated Interrupts (vSGI) by asserting the ICH_HCR_EL2.vSGIEOICOUNT bit. SGI interrupt identifications are from 15 to 0, however the masking inhibits interrupt identifications from 31 to 0, which includes some virtual Private Peripheral Interrupt identifications.

Configurations Affected

The erratum affects all configurations when GICCDISABLE pin is not tied to '1'.

Conditions

1. ICH_HCR_EL2.vSGIEOICOUNT is asserted.
2. Virtual interrupt identification between 16 and 31 is deactivated on the virtual interface.

Implications

If the above conditions are met, then ICH_HCR_EL2.EOIcount will not advance for some virtual Private Peripheral Interrupt identifications.

Workaround

Do not set ICH_HCR_EL2.vSGIEOICOUNT and rely on the GICv4.0 architecture counting methods.

1618630
Branch prediction for an ERET cached in the instruction cache might cause a deadlock

Status

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

Description

When a branch predictor makes a prediction for an ERET instruction, the core might deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core executes a conditional branch instruction.
2. The branch predictor caches the branch in Condition 1.
3. The branch instruction is overwritten by an ERET instruction by a self-modifying code sequence.
4. The core caches the ERET instruction in the instruction cache, and later fetches the ERET instruction from the cache.
5. The branch predictor makes a prediction for the ERET based on the branch information cached at Condition 2.
6. The predicted target matches ELR[PSTATE.EL].

Implications

If the above conditions are met, then the core might deadlock.

Workaround

Instruction patching, through hardware registers, for an ERET instruction prevents ERET instructions from entering into this scenario. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers:

```
LDR x0,=0x7
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0xF3D08000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFF0F0FF
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x80000002003FF
```

```
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
ISB
```

1618634
Incorrect instructions might be executed

Status

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

Description

Incorrect instructions might be executed in AArch64 state.

Configurations Affected

This erratum affects all configurations with CORE_POP_RAM set to TRUE.

Conditions

1. The core executes in AArch64 state.
2. A specific sequence of L0 and L1 instruction cache misses occur.
3. A table walk response arrives at the L1 instruction TLB at the same time a lookup occurs, and the lookup instruction address overlaps the incoming page mapping.

Implications

If the above conditions are met, then the core might execute incorrect instructions.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1 to disable a performance feature. This should be done before enabling the MMU.

1618635
NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock

Status

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

Description

Under certain conditions, execution of either a load to device or non-cacheable memory, and either a store exclusive or register read of the PAR (physical address register) in close proximity might lead to a deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Execution of any load with device or non-cacheable memory attributes, and
2. Execution of a store-exclusive or register read of PAR.

Implications

If the above conditions are met, then the core might stop executing code.

Workaround

This issue can be worked around by using the instruction patching mechanism. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers. The code sequence should be applied early in the boot sequence prior to any of the possible errata conditions being met.

```
;; Inserts a DMB SY before and after MRS PAR_EL1
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,= 0xEE070F14
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,= 0xFFFF0FFF
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x4005027FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0

;; Inserts a DMB SY before STREX imm offset
LDR x0,=0x1
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8400000
MSR S3_6_c15_c8_2,x0
```

```
LDR x0,=0x00fff00000
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4001027FF
MSR S3_6_c15_c8_1,x0

;; Inserts a DMB SY before STREX[BHD}/STLEX*
LDR x0,=0x2
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8c00040
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00fff00040
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4001027FF
MSR S3_6_c15_c8_1,x0

;; Inserts a DMB SY after STREX imm offset
LDR x0,=0x3
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8400000
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00fff00000
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4004027FF
MSR S3_6_c15_c8_1,x0

;; Inserts a DMB SY after STREX[BHD}/STLEX*
LDR x0,=0x4
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8c00040
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00fff00040
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4004027FF
MSR S3_6_c15_c8_1,x0

;; Synchronize to enable patches
ISB
```

1618636
Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core

Status

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

Description

Under certain conditions, when a Floating Point (FP)/Advanced SIMD instruction is attempting to dispatch but is flushed due to a mispredicted branch, a correct-path Aarch32 Conditional FP/Advanced SIMD instruction might fail to schedule for execution, resulting in a deadlock in the core.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A long period with no FP/Advanced SIMD instruction activity, followed by
2. A flag writing instruction, and
3. A mispredicted branch, with an FP/Advanced SIMD instruction on the mispredicted path and an Aarch32 Conditional FP/Advanced SIMD instruction on the correct path.

Implications

If the above conditions are met, then this erratum might result in a hang.

Workaround

The workaround is to set CPUACTLR5_EL1[8] to 1'b1. The workaround might result in a small increase in core power consumption.

1619807
The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache

Status

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

Description

When the core executes an instruction during an active-not-pending state in a software step or halt step process, the core might execute multiple instructions before taking software step exception or halt step exception.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Software step or halt step is enabled in the AArch64 instruction state.
2. Instruction fetch hits in the L0 Macro-op cache.

Implications

If the above conditions are met, then the core might execute multiple instructions before taking a software step exception or halt step exception.

Workaround

Set CPUACTLR_EL1[11] to one, which flushes the L0 Macro-op cache for all context synchronization events.

1619814
Enabling SPE might result in deadlock in some situations

Status

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

Description

Use of SPE might result in a deadlock in some situations.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A Floating-point Divide or Floating-point Square Root instruction gets dispatched.
2. This instruction gets flushed.
3. A Vector Unit instruction gets sampled by SPE post flush.
4. A DVM Sync gets issued subsequently.

Implications

If the above conditions are met, then the completion tracker for the SPE sample does not progress, which might prevent any DVM Sync issued subsequently from completing and cause a deadlock.

Workaround

This erratum can be avoided by disabling SPE, by setting PMBLIMITR_EL1.E = 0. However, the deadlock is found to occur rarely, therefore SPE could be enabled and used for prototyping purposes.

1654562
A streaming write in the presence of a store-release instruction might result in data corruption

Status

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

Description

Under certain micro-architectural conditions, a streaming write in the presence of a store-release instruction might result in data corruption.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A streaming store, with address A, is executed.
2. A store-release instruction, with address B, is dispatched before it is the oldest. However, the write is cancelled and retried to maintain ordering.
3. A subsequent cacheable, non-streaming store with address C is executed next.

Implications

If the above conditions are met under certain micro-architectural conditions, then this erratum might result in data corruption.

Workaround

This erratum can be avoided by setting CPUACTLR2_EL1[1] to 1, which prevents the store-release from being dispatched before it is the oldest.

1659792
Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation

Status

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

Description

When Stage 2 dirty and access flag updates are turned off, a failed profiling buffer translation request might result in reporting a Stage 2 Data Abort code in PMBSR_EL1.EC. This also results in an Unsupported Exclusive or Atomic Access fault status code update in PMBSR_EL1, which is not one of the defined FSC codes for this register.

Configurations Affected

This erratum affects all configurations.

Conditions

SPE is enabled and the following conditions are true:

1. Hardware Management of dirty state and access flag update in Stage 1 translations is enabled in TCR_EL1.
2. Hardware Management of dirty state and access flag update in Stage 2 translations is disabled.

Implications

There might be a loss of sampling data as software needs to restart the profiling session to recover from this error.

Workaround

This erratum can be avoided by pre-dirtying the SPE buffer pages.

1659794
Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer

Status

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

Description

A profiling buffer translation request might speculatively update the translation table descriptor of the page following the Statistical Profiling Buffer. If dirty bit management is enabled, then this request might result in setting the dirty bit.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A buffer full event is signaled coincident to the sampling interval running down to 0, causing a sampling pulse, following the last valid record write.
2. No other transactions access the virtual address page following the Profiling Buffer.

Implications

If the above conditions are met, then the sample that is initiated coincident to the buffer full indicator, forces a translation request for the new buffer page, which might result in a table walk and update the translation table descriptor.

Workaround

This erratum can be avoided by mapping and reserving a writable virtual address page at the end of the Profiling Buffer.

1674403
The core might deadlock when executing a program sequence with divide or square-root operations

Status

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

Description

Under certain micro-architectural conditions, a program sequence with divide or square-root operations along with a mispredicted branch can result in a deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Multiple Floating-point/Advanced SIMD operations, including multiple divide or square-root operations are executed.
2. Branch instruction mispredicts.
3. Subsequent divide or square-root operation is executed.

Implications

If the above conditions are met under certain micro-architectural conditions, then this erratum might result in a deadlock.

Workaround

To avoid this erratum, set CPUACTLR3_EL1[12] to 1. This disables result bus sharing in the Floating-point/Advanced SIMD unit. The overall performance loss on SPECfp workloads is less than 0.5%.

1774420
A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB

Status

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

Description

Under certain conditions, a transient single-bit ECC error in the MMU TC RAM might prevent a TLB invalidate (TLBI) instruction from removing the entry. If the transient error is not detected for a subsequent miss request targeting the affected page, then the MMU might return a stale translation.

Configurations affected

All configurations are affected.

Conditions

All of the following conditions must be met:

1. Both stage 1 and stage 2 translations are enabled.
2. Stage 1 page or block size is larger than stage 2 page or block size.
3. MMU TC RAM entry has a transient single-bit ECC error.
4. TLBI targets the translation in the MMU TC RAM entry containing the single-bit ECC error.
5. The single-bit ECC error prevents the TLBI from removing the entry.
6. Transient single-bit ECC error goes away before a subsequent translation request matching the L2 TLB entry is issued.

Implications

If the above conditions are met, then the MMU might return stale translation for a subsequent access. The transient single-bit ECC error will be reported in ERR0MISC0_EL1 register.

Workaround

This erratum can be avoided by setting CPUECTLR_EL1[53] to 1, which disables the allocation of splintered pages in the L2 TLB.

1791573
Atomic Store instructions to shareable write-back memory might cause memory consistency failures

Status

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

Description

Atomic Store instructions to shareable write-back memory that are performed as far atomics might cause memory consistency failures if the initiating PE has a shared copy of the cache line containing the addressed memory.

Configurations affected

This erratum affects all configurations that have an interconnect capable of handling far atomic transactions indicated by the BROADCASTATOMIC pin being set to 1.

Conditions

1. PE0 executes Atomic Store instruction that hits in the L1 data cache and L2 cache in the Shared state.
2. PE0 changes the L2 state to Invalid, sends an invalidating snoop to the L1 data cache, and issues a AtomicStore transaction on the CHI interconnect.
3. PE0 invalidating snoop to the L1 data cache is delayed due to internal queueing.

Implications

If the above conditions are met, PE0 might not observe invalidating snoops caused by other PEs in the same coherency domain and thus might violate memory consistency for loads to the same cache line as the Atomic Store.

Workaround

Set CPUACTLR2_EL1[2] to force Atomic Store operations to write-back memory to be performed in the L1 data cache.

1852267
SPE might illegally write to any physical address, including Secure space

Status

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

Description

Under certain conditions, SPE requests may use the VA as a PA and therefore may write any physical memory location, including Secure space.

Configurations Affected

This erratum affects all configurations with RANDOM_NUMBER set to TRUE.

Conditions

SPE is enabled and the following conditions are true:

1. MRS RNDR or MRS RNDRRS has executed. This could be on the speculative path.
2. The SPE buffer generates a translation request.

Implications

If the above conditions are met, the SPE may use the VA as a PA and therefore may write any physical memory location, including Secure space.

Workaround

To work around this erratum, EL3 software at boot time should set CPUACTLR2[28] to disable SPE sampling.

1863563
Core might generate breakpoint exception on incorrect IA

Status

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

Description

Under certain rare conditions, the core can generate a breakpoint exception on the instruction that is sequentially before the address specified in DBGBVR<n>_EL1.

Configurations Affected

This erratum affects all configurations.

Conditions

This exception might occur when:

1. Hardware breakpoint is enabled.
2. CPU instruction execution is not being single stepped.

Implications

If the above conditions are met, a breakpoint exception programmed for a given PC might instead cause a breakpoint exception for the instruction at PC-4.

Workaround

If software recognizes that a breakpoint exception has occurred for PC-4, when a breakpoint was expected at PC, then an instruction step should be performed.

Note: this erratum was previously published with a different workaround, which entailed setting CPUACTLR_EL1[21] to 1'b1. That workaround should only be applied to r0p0 hardware.

1915893
Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled.

Status

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

Description

Virtual address (VA) to physical address (PA) translation latency is not captured in SPE records when physical address collection is disabled at the appropriate exception level (EL).

Configurations Affected

This erratum affects all configurations.

Conditions

1. Physical address collection is disabled for SPE records at the appropriate EL by setting PMSCR_EL1.PA=0 or PMSCR_EL2.PA=0.

Implications

If the above conditions are met, then the translation latency value is not captured in the SPE records.

Workaround

Where it is acceptable to capture the physical address, this erratum can be avoided by enabling physical address sampling, by setting PMSCR_EL1.PA = 1 and PMSCR_EL2.PA = 1.

1923200
External debugger access to Debug registers might not work during Warm reset

Status

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

Description

During Warm reset, external debugger access for Debug registers might be ignored.

Configurations Affected

All configurations are affected.

Conditions

1. Warm reset is asserted.
2. External debugger access is initiated for one of following Debug registers:
    - DBGBCR<n>_EL1 (n=0-5)
    - DBGBVR<n>_EL1 (n=0-5)
    - EDECCR

Implications

If the above conditions are met, the core might ignore the access request. The read operation might return incorrect data. The write operation might not take effect and stale data might be retained.

Workaround

There is no workaround.

1925756
Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state

Status

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

Description

Under certain circumstances, a store operation that encounters multiple hits in the TLB can generate a prefetch request to regions of memory with attributes that could not be accessed at that Exception level or Security state.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A store operation encounters multiple hits in the TLB due to inappropriate invalidation or misprogramming of a contiguous bit.
2. A read request is generated with a physical address and attributes that are an amalgamation of the multiple TLB entries that hit.

Implications

If the above conditions are met, a read request might be generated to regions of memory with attributes that could not be accessed at that Exception level or Security state. The memory location will not be updated.

Workaround

This erratum can be avoided by setting CPUECTLR_EL1[8] to 1. There is a small performance cost (<0.5%) for setting this bit.

1940577
Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics

Status

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

Description

Under certain conditions, atomic instructions with acquire semantics might not be ordered with respect to older instructions with release semantics. The older instruction could either be a store or store atomic.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Load atomic, CAS, or SWP with acquire but no release semantics is executed.
2. There is an older instruction with release semantics and it could either be a store to non-WB memory or a store atomic instruction that is executed as a far atomic.

Implications

If the above condition are met, memory ordering violation might happen.

Workaround

This erratum can be avoided by inserting a DMB ST before acquire atomic instructions without release semantics. On r1p0 or r1p1 hardware, this can be implemented through execution of the following code at EL3 as soon as possible after boot:

```
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,= 0x10E3900002
MSR S3_6_c15_c8_2,x0
LDR x0,= 0x10FFF00083
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x2001003FF
MSR S3_6_c15_c8_1,x0
```

```
LDR x0,=0x1
MSR S3_6_c15_c8_0,x0
LDR x0,= 0x10E3800082
MSR S3_6_c15_c8_2,x0
LDR x0,= 0x10FFF00083
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x2001003FF
MSR S3_6_c15_c8_1,x0

LDR x0,=0x2
MSR S3_6_c15_c8_0,x0
LDR x0,= 0x10E3800200
MSR S3_6_c15_c8_2,x0
LDR x0,= 0x10FFF003E0
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x2001003FF
MSR S3_6_c15_c8_1,x0

ISB
```

Note that there is no workaround provided for r0p0 hardware. Please contact Arm support for further details.

1966096
The instruction following a change in SVE Vector length in ZCR_EL1 might not see updated value

Status

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

Description

Instructions following a change to the vector length via MSR ZCR_EL1 are required to see that new vector length, for example, VL128 or VL256, without need for explicit context synchronization. Under some conditions the instruction immediately following MSR ZCR_EL1 will not observe the updated value.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Mop Cache is enabled.
2. MSR ZCR_EL1 executes, changing the vector length.
3. An immediately subsequent vector length consuming instruction executes, such as RDVL.

Implications

The instruction immediately following MSR ZCR_EL1 might use the incorrect vector length for its calculation.

Workaround

This erratum can be avoided on r1p0 or r1p1 hardware through execution of the following code at EL3 as soon as possible after boot:

```
LDR x0,=0x3
MSR S3_6_c15_c8_0,x0
LDR x0,= 0xEE010F12
MSR S3_6_c15_c8_2,x0
LDR x0,= 0xFFFF0FFF
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x80000000003FF
MSR S3_6_c15_c8_1,x0

ISB
```

Note that there is no workaround provided for r0p0 hardware. Please contact Arm support for further details.

1978082
Incorrect programming of PMBPTR_EL1 might result in a deadlock

Status

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

Description

When PMBPTR_EL1 is incorrectly programmed to be equal to or greater than PMBLIMITR_EL1, then under certain conditions, the CPU might deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. SPE is enabled.
2. PMBSR_EL1.S = 0, indicating PMBIRQ is not asserted.
3. PMBPTR_EL1 is programmed to be equal to or greater than PMBLIMITR_EL1.

Implications

If the above conditions are met, then the CPU might deadlock. Note that software written correctly will not expose this erratum.

Workaround

This erratum can be avoided by mediating access to the SPE control registers from a higher exception level.

A hypervisor at EL2 can configure MDCR_EL2.E2PB to trap EL1 accesses to PMBPTR_EL1, PMBLIMITR_EL1, and PMBSR_EL1. The hypervisor can mediate these accesses and maintain a shadow copy of PMBLIMITR_EL1 such that the physical PMBLIMITR_EL1 register has PMBLIMITR_EL1.E clear whenever PMBPTR_EL1.PTR >= PMBLIMITR_EL1.LIMIT.

Firmware at EL3 can configure MDCR_EL3.NSPB to disable SPE in the active security state and trap erroneous EL1/EL2 accesses to the SPE registers. Software written correctly should not access the SPE registers in this case.

2108267
Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock

Status

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

Description

If the data prefetcher is disabled (by an MSR to CPUECTLR register) while a prefetch TLB miss is outstanding, the processor might deadlock on the next context switch.

Configurations Affected

All configurations are affected.

Conditions

- MSR write to CPUECTLR register that disables the data prefetcher.
- A TLB miss from the prefetch TLB is outstanding.

Implications

If the above conditions are met, a deadlock might occur on the next context switch.

Workaround

- Workaround option 1:
  If the following code surrounds the MSR, it will prevent the erratum from happening:
  - tlbi (to blind address) local version (does not have to be broadcast)
  - dsb
  - isb
  - MSR CPUECTLR - disabling the prefetcher
  - isb

- Workaround option 2:
  Place the data prefetcher in the most conservative mode instead of disabling it. This will greatly reduce prefetches but not eliminate them. This is accomplished by writing the following bits to the value indicated:
  - ecltr[7:6], PF_MODE = 2'b11

2139242
Hardware data prefetcher can generate a physical address with incorrect attributes

Status

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

Description

Under some conditions, the hardware data prefetcher can generate a physical address request to the memory system with cacheable attribute, when the access is actually NC or DEV.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Hardware data prefetcher is enabled.
2. Unaligned load is executed in a 4k translation page size without crossing a page boundary, the unaligned load is to cacheable memory.
3. Mixed translation page sizes for data are being used, one of the pages is 4k in size, one of the pages is 64k or greater.
4. One of the RNDR instructions (MRS X(t) RNDR, MRS X(t) RNDRRS) is executed, or one of the v8.4-NV Enhanced Support for Nested Virtualization Operations (loads and stores generated by transforming register accesses) is executed and the effective value of HCR_EL2.NV2 is 0x1, in the same ELx as the unaligned load.

Implications

A physical address request might be generated by the hardware data prefetcher with incorrect attributes and presented to memory subsystem. This physical address will be to a location in a 64 block of physical memory, and is the same block of physical memory that contains the physical address location that the unaligned load is addressing.
The attributes used by the prefetch address will be the same attributes as the unaligned load address, and might not match the attributes of the physical location being accessed. For example, the attribute of prefetch address might be cacheable, while the particular physical 4k page might be mapped NC.

Workaround

This erratum can be avoided by implementing one of the following workaround options:

1. Do not use the RNDR function and disable enhanced support for nested virtualization by setting HCR_EL2.NV2 to 0.
2. Disable support for nested virtualization and program the following instruction patch to serialize before and after each RNDR[RS] instruction. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers, early in the boot sequence:

```
LDR x0,=0x4
 MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
 LDR x0,=0xEE720F14
 MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
 LDR x0,=0xFFFF0FDF
 MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
 LDR x0,=0x40000005003FF
 MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
 ISB
```

   Note that there is no workaround provided for r0p0 hardware. Please contact Arm support for further details.

2216392
PDP deadlock due to CMP/CMN + B.AL/B.NV fusion

Status

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

Description

When Performance Defined Power (PDP) is enabled, a Compare (CMP) or Compare negative (CMN) instruction followed by a conditional branch of form B.AL or B.NV might cause a deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. PDP configuration is enabled.
2. Execution of CMP/CMN, followed by B.AL/B.NV.

Implications

If above conditions are met, then a deadlock might result, requiring a reset of the processor.

Workaround

This erratum can be avoided by applying following patch. These instructions are not expected to be present in the code often, so any performance impact should be minimal. The code sequence should be applied early in the boot sequence prior to any of the possible errata conditions being met.

```
LDR x0,=0x5
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0x10F600E000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0x10FF80E000
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x80000000003FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0

ISB
```

Note that there is no workaround provided for r0p0 hardware. Please contact Arm support for further details.

2294912
Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch

Status

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

Description

A PE executing a PLDW or PRFM PST instruction that lies on a mispredicted branch path might cause a second PE executing a store exclusive to the same cache line address to fail continuously.

Configurations Affected

This erratum affects all configurations.

Conditions

1. One PE is executing store exclusive.
2. A second PE has branches that are consistently mispredicted.
3. The second PE instruction stream contains a PLDW or PRFM PST instruction on the mispredicted path that accesses the same cache line address as the store exclusive executed by the first PE.
4. PLDW/PRFM PST causes an invalidation of the first PE's caches and a loss of the exclusive monitor.

Implications

If the above conditions are met, the store exclusive instruction might continuously fail.

Workaround

Set CPUACTLR2_EL1[0] to 1 to force PLDW/PFRM ST to behave like PLD/PRFM LD and not cause invalidations to other PE caches. There might be a small performance degradation to this workaround for certain workloads that share data.

2372203
Translation table walk folding into an L1 prefetch might cause data corruption

Status

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

Description

A translation table walk that matches an existing L1 prefetch with a read request outstanding on CHI might fold into the prefetch, which might lead to data corruption for a future instruction fetch.

Configurations Affected

This erratum affects all configurations

Conditions

1. In specific microarchitectural situations, the PE merges a translation table walk request with an older hardware or software prefetch L2 cache miss request.

Implications

If the previous conditions are met, an unrelated instruction fetch might observe incorrect data.

Workaround

Disable folding of demand requests into older prefetches with L2 miss requests outstanding by setting CPUACTLR2_EL1[40] to 1.

2675258
Static and dynamic TXREQ limiting might cause deadlock

Status

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

Description

Use of the static and dynamic TXREQ limiting functions might cause a system deadlock. These functions are disabled by default.

Configurations Affected

This erratum affects all system configurations that include a component that can create a forward progress dependency on a older transaction through new transactions. Such components include the Chip-to-Chip Gateway block of CMN interconnect and PCIe Root Complexes.

Conditions

Under specific conditions involving request traffic to the specified components, the static and dynamic TXREQ limiting function might prevent a retried transaction from making forward progress.

Implications

If the above conditions are met, a retried CHI request might never be reissued, potentially leading to a system deadlock.

Workaround

Do not enable static or dynamic TXREQ limiting functions by keeping CPUECTLR2_EL1[2] at 0b0 and CPUECTLR2_EL1[1:0] at 0b00. These are the reset values.

2701953
The core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back

Status

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

Description

If a core is fetching instructions from memory while stage 1 translation is disabled and instruction cache is disabled, the core ignores Stage 2 forced Write-Back indication programmed by HCR_EL2.FWB and make Non-cacheable, Normal memory request. This may cause the core to fetch stale data from memory subsystem.

Configurations Affected

This erratum might affect system configurations that do not use Arm interconnect IP.

Conditions

The erratum occurs if all the following conditions apply:

- The *Processing Element* (PE) is using EL1 translation regime.
- Stage 2 translation is enabled (HCR_EL2.VM=1).
- Stage 1 translation is disabled (SCTLR_EL1.M=0).
- Instruction cache is enabled from EL2 (HCR_EL2.ID=0).
- Instruction cache is disabled from EL1 (SCTLR_EL1.I=0).

Implications

If the conditions are satisfied, the core makes all instruction fetch request as Non-cacheable, Normal memory regardless of stage 2 translation output even if Stage 2 Forced Write-back is enabled. This might cause the core to fetch stale data from memory because Non-cacheable memory access does not probe any of cache hierarchy (e.g., Level-2 cache). If the bypassed cache hierarchy contains data modified by other initiators, stale data might be fetched from memory.

Workaround

For Hypervisor, initiating appropriate cache maintenance operations as if the core does not support stage 2 Forced Write-back feature. The cache maintenance operation should be initiated when new memory is allocated to a guest OS. This operation writeback the modified data in intermediate caches to point of coherency.

2743093
The core might deadlock during powerdown sequence

Status

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

Description

While powering down the *Processing Element* (PE), a correctable L2 tag ECC error might cause a deadlock in the powerdown sequence.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. Error detection and correction is enabled through ERXCTLR_EL1.ED=1.
2. PE executes more than 24 writes to Device-nGnRnE or Device-nGnRE memory.
3. PE executes powerdown sequence as described in the *Technical Reference Manual* (TRM).

Implications

If the above conditions are met, the PE might deadlock during the hardware cache flush that automatically occurs as part of the powerdown sequence.

Workaround

Add a DSB instruction before the ISB of the powerdown code sequence specified in the TRM.

2743233
Page crossing access that generates an MMU fault on the second page could result in a livelock

Status

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

Description

Under unusual micro-architectural conditions, a page crossing access that generates a *Memory Management Unit* (MMU) fault on the second page can result in a livelock.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under all of the following conditions:

1. Page crossing load or store misses in the *Translation Lookaside Buffer* (TLB) and needs a translation table walk for both pages.
2. The table walk for the second page results in an MMU fault.

Implications

If the above conditions are met, under unusual micro-architectural conditions with just the right timing, the core could enter a livelock. This is expected to be very rare and even a slight perturbation due to external events like snoops could get the core out of livelock.

Workaround

This erratum can be avoided by setting CPUACTLR5_EL1[56:55] to 2'b01.

2779461
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, r1p0, r1p1, r1p2. Open.

Description

The Processing Element (PE) might generate memory accesses using invalidated mappings after completion of a Distributed Virtual Memory (DVM) SYNC operation.

Configurations Affected

All configurations are affected.

Conditions

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

Implications

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

Workaround

The erratum can be avoided by setting CPUACTLR3_EL1[47]. Setting this chicken bit might have a small impact on power and negligible impact on performance.

3003013

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations with double fault extension.

Conditions

This erratum occurs under the following conditions:

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

Implications

The DRPS instruction is not executed correctly.

Workaround

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

3028884
SPE might write to pages which lack write permission at Stage-1 or Stage-2

Status

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

Description

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

Configurations affected

This erratum affects all configurations that support SPE.

Conditions

This erratum occurs under the following conditions:

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

Implications

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

Workaround

No hardware workaround is available.

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

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

No current software is expected to have this problem.

3324341 MSR PSTATE.SSBS to 0 is not fully self-synchronizing

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following condition applies:

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

Implications

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

Workaround

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

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

3696285
Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

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

or

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

Implications

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

Workaround

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

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

Category B (rare)

1543964
The core might fetch a stale instruction from memory which violates the ordering of instruction fetches

Status

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

Description

When the core executes a direct branch that has been recently modified and is associated with prefetch speculation protection, the core might fetch a stale instruction which violates the ordering of instruction fetches.

Configurations Affected

This erratum affects all multi-core configurations.

Conditions

1. The modifying core changes instructions at address, A.
2. The modifying core executes cache-maintenance and synchronization instructions to make address A visible to all cores in the inner shareable domain.
3. A direct branch or a NOP is substituted with a direct branch targeting address A on the modifying core.
4. The executing core fetches the branch and correctly predicts the destination of the direct branch based on stale history due to Address Space Identifier (ASID) or Virtual Machine Identifier (VMID) reuse.
5. Stale instructions are fetched from the L0 Macro-op cache, on the executing core, instead of the modified instructions at address A.

Implications

Software relying on prefetch speculation protection, instead of explicit synchronization when modifying a direct branch, might execute stale instructions when the branch is taken.

Workaround

To avoid this erratum, invalidate the branch history before reusing any ASID for a new address space. This can be done by ensuring 124 ASIDs are selected before any ASID is re-used.

1618631
MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data

Status

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

Description

An MRRC read of certain Generic Timer system registers in AArch32 mode might return corrupt data.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met under rare internal timing conditions:

1. The core is executing at AArch32 at EL0.
2. An MRRC to CNTPCT, CNTVCT, CNTP_CVAL, or CNTV_CVAL is executed.

Implications

If the erratum occurs, then the second destination register [Rt2] of the MRRC will incorrectly contain the same data as the first destination register [Rt].

Workarounds

The erratum can be avoided by trapping MRC/MCR/MRRC/MCRR accesses in AArch32 to the affected registers and doing the equivalent code sequence in the trap handler.
To trap the CNT* accesses, set CNTKCTL_EL1.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0. If HCR_EL2.{E2H,TGE}={1,1} then set CNTHCTL_EL2.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0.
The following registers will be trapped:

- CNTP_CTL.
- CNTP_CVAL.
- CNTP_TVAL.
- CNTV_CTL.
- CNTV_CVAL.
- CNTV_TVAL.

- CNTPCT.
- CNTVCT.
- CNTFRQ.

2348377
AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment

Status

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

Description

AMBA CHI protocol credit grants might not be distributed in a fair manner inside the *Processing Element* (PE), leading to a livelock condition.

Configurations Affected

This erratum affects all configurations.

Conditions

Under specific microarchitectural conditions and the presence of protocol retries on the AMBA CHI interface, the PE might fail to assign protocol credits in a fair manner across L2 cache banks.

Implications

If the above conditions are met, a retried CHI request might never consume a protocol credit provided by a PCrdGrant response and will therefore not be re-issued to the interconnect, potentially leading to a livelock of the PE.

Workaround

Set CPUACTLR5_EL1[61] to 1.

2986644
PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level

Status

Fault Type: Programmer Category B (Rare)
Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

Description

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

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

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

- If MDCR_EL2.TDE == 0, a stage 2 Data Abort might result in a Data Abort exception taken erroneously to EL1.
  - The rarity of PE internal timings required to exhibit this bug is comparable to *Reliability, Availability, and Serviceability* (RAS) error FIT rates. Expected outcome is a kernel panic that will kill the process.

- If MDCR_EL2.TDE == 1, a stage 1 Data Abort might result in a Data Abort exception taken erroneously to EL2.
  - This scenario is containable within a hypervisor via the software workaround outlined below.

Workaround

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

EL2 can protect against this case as follows:

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

Category C

1452392
Unaligned SVE Non-Fault load that encounters a tag double-bit error or data poison might resolve with an abort

Status

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

Description

The first half of an unaligned SVE Non-Fault load that encounters either an L1 tag double-bit error or data poison might result in the load resolving with an abort instead of updating the First Fault Register.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An SVE Non-Fault load either crosses a cache line boundary or is of size 32 bytes and not 4-byte aligned.
2. While this load is executing down the load/store pipeline, it becomes the oldest unresolved instruction.
3. The first half of the Non-Fault load encounters either an L1 tag double-bit error or data poison before it is identified as the oldest unresolved instruction.

Implications

If the above conditions are met, then under specific microarchitectural timing conditions, SVE Non-Fault loads might resolve with an abort instead of updating the First Fault Register.

Workaround

There is no workaround for this erratum.

1507873
Misaligned atomics to unsupported Non-cacheable or Device memory location might lead to incorrect DFSC code in the ESR

Status

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

Description

If an Armv8.4-A atomic access to Non-cacheable or Device memory is misaligned but all bytes of the memory access lie within a 16-byte quantity aligned to 16 bytes, then the atomic access might cause an alignment fault. If the system does not support the atomic access to Non-cacheable or Device memory regions, then the atomic access might also trigger an IMPLEMENTATION DEFINED abort, however the alignment fault should take priority. In a virtualized environment, if a Stage 1 translation generates a Write-Back memory attribute but the final memory type of the access is Non-cacheable or Device, then the core might incorrectly report an unsupported atomic fault.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An Armv8.4-A atomic access is misaligned but all bytes of the memory access lie within a 16-byte quantity aligned to 16 bytes.
2. The atomic access is to a Non-cacheable or Device memory location and the system does not support it.
3. Virtualization is enabled and the access occurs at EL1.
4. A Stage 1 translation for the access maps it to a Write-Back memory region and Stage 2 mappings generate a final Non-cacheable or Device memory type.

Implications

If the above conditions are met, then an exception is taken with an incorrect DFSC code in the ESR indicating an IMPLEMENTATION DEFINED unsupported atomic fault.

Workaround

There is no workaround for this erratum.

1549423
Transient parity error in L1 instruction cache might result in missed breakpoint exception

Status

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

Description

When a transient parity error occurs in the L1 instruction cache close to an address breakpoint, then under certain conditions the core might ignore the breakpoint.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The breakpoint is set on a cacheable line.
3. A transient parity error occurs when reading the L1 instruction cache near the breakpoint location.
4. At least one RAMINDEX operation targeting the L1 instruction cache in the core with the breakpoint is outstanding.

Implications

If the above conditions are met, then the core might ignore the address breakpoint.

Workaround

Use a synchronization instruction, such as ISB, with the RAMINDEX functionality.

1553581
ID_PFR1_EL1.GIC incorrectly reports as 0x1

Status

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

Description

The value of ID_PFR1_EL1.GIC incorrectly reports a value of 0x1. This value corresponds to GIC CPU Interface v3.0/v4.0. The correct value is 0x2 which corresponds to GIC CPU Interface v4.1.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An MRS to ID_PFR1_EL1 is executed.

Implications

If the above conditions are met, then the ID field ID_PFR1_EL1.GIC will be reported incorrectly.

Workaround

This issue can be worked around in software by ignoring the value of 0x1 in ID_PFR1_EL1.GIC and assuming the correct value of 0x2 based on the MIDR_EL1 of the affected part.

1591505
Predicate event information populated incorrectly for non-predicated SVE instructions sampled by SPE

Status

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

Description

Under specific conditions, certain non-predicated Scalable Vector Extension (SVE) instructions, when sampled by Statistical Profiling Extension (SPE), report partial and empty predicate events in the SPE record.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A non-predicated SVE instruction gets sampled by SPE.
2. This instruction either has a partial or empty predicate.

Implications

If the above conditions are met, then the partial and empty events for the corresponding SPE record are set based on the predicate registers.

Workaround

This erratum can be avoided by filtering out "Other" SPE samples.

1595153 ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect

Status

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

Description

Under certain conditions, the ERR0MISC0_EL1.SUBARRAY value recorded for ECC errors in the L1 data cache might be incorrect.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A load, store, or atomic instruction accesses multiple banks of the L1 data cache.
2. One of the banks accessed has an ECC error.

Implications

If the above conditions are met, then ERR0MISC0_EL1.SUBARRAY might have an incorrect value. The remaining fields of the ERR0MISC0_EL1 register remain correct.

Workaround

There is no workaround for this erratum.

1596291
Data side MPAM values used for PLI

Status

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

Description

A memory-system request is generated as a result of a PLI instruction with data side MPAM values.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Processing Element (PE) executes PLI instruction.
2. A memory-system request is generated as a result of PLI instruction.

Implications

If the above conditions are met, then the PE generates a memory-system request for instruction access with data side MPAM information.

Workaround

No workaround required.

1618632
TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0

Status

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

Description

TLBI instructions are not treating ASID[15:8] as zero when TCR_EL1.AS=0, as specified in the Arm Architecture Reference Manual. In this configuration, the bits are RES0, which should be written to zero by software, and ignored by hardware.

Configurations Affected

The erratum affects all configurations.

Conditions

1. TCR_EL1.AS=0.
2. A TLBI is executed with ASID[15:8] not equal to zero.

Implications

The TLBI executes locally and broadcasts with an ASID that is out of range for this configuration.

Workaround

This erratum can be avoided if software is properly writing zero to RES0 bits.

1618633
Waypoints from previous session might cause single-shot comparator match when trace enabled

Status

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

Description

On the first waypoint after the core ETM is enabled, it is possible for a single-shot comparator to have a spurious match based on the address from the last waypoint in the previous trace session.

Configurations Affected

This erratum affects all configurations.

Conditions

- The core ETM has been enabled, disabled, and re-enabled since the last reset.
- Single-shot address comparators are enabled.
- The last waypoint address before the core ETM was disabled either matches a single-shot comparator or causes a match in the range between waypoints depending on the single-shot control setup.

Implications

There might be a spurious single-shot comparator match, which might be used by the trace analyzer to activate other trace events.

Workaround

Between tracing sessions, set the core ETM to enter a prohibited region either instead of or in addition to disabling the ETM.

1618637
Interrupt might be taken later than architecturally mandated on exit from Debug state

Status

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

Description

An interrupt might not be taken before the first instruction after Debug state exit if all of the following is true:

- The interrupt becomes pending during Debug state.
- The interrupt becomes unmasked by explicitly clearing interrupt mask bits in DSPSR_EL0 before Debug state exit, such that when DSPSR_EL0 is copied to PSTATE on Debug state exit the interrupt mask bits are cleared.
- A change in Execution state is involved on Debug state exit.

Configurations Affected

This erratum affects all configurations.

Conditions:

Case A:

1. Enter Debug state from AArch64 with an interrupt masked (PSTATE.A==1 | PSTATE.I==1 | PSTATE.F==1).
2. While in Debug state, execute multiple FMOV instructions that write to vector registers of a size less than a quadword.
3. While in Debug state, execute an MSR DPSR_EL0 to stipulate a return with an Execution state change (to AArch32) and with an interrupt unmasked (PSTATE.A==0 | PSTATE.I==0 | PSTATE.F==0).
4. Exit Debug state with an interrupt pending.

Case B:

1. Enter Debug state from AArch32 with an interrupt masked (PSTATE.A==1 | PSTATE.I==1 | PSTATE.F==1).
2. While in Debug state, execute a DCPSx instruction to move to a higher EL (switching Execution state to AArch64).
3. While in Debug state, execute multiple FMOV instructions that write to vector registers of a size less than a quadword.
4. While in Debug state, execute an MSR DPSR_EL0 to stipulate a return with an Execution state

change (to AArch32) and with an interrupt unmasked (PSTATE.A==0 | PSTATE.I==0 | PSTATE.F==0).
5. Exit Debug state with an interrupt pending.

Implications

The interrupt will be recognized, but may not be recognized before the first instruction after Debug state exit. In cases where interrupt recognition is late, it will occur before the second instruction after Debug state exit.

Workaround

No workaround is suggested for this erratum, because it is not expected that this erratum will be encountered by systems under normal operating conditions and the implications of late interrupt recognition under these circumstances are not considered harmful.

1619800
An unaligned load may initiate a prefetch request which crosses a page boundary

Status

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

Description

A load which crosses a 64-byte boundary, but not a 4KB boundary, and hits a TLB entry for a page which is less than 64KB in size, might trigger a prefetch request which incorrectly interprets the page size to be 64KB and therefore initiates a read request for an unexpected physical address.

Configurations Affected

The erratum affects all configurations.

Conditions

1. The system is configured with read-sensitive Device memory at a physical address which overlaps with an aligned 64KB region that belongs to Normal memory.
2. A load which crosses a 64-byte boundary, but not a 4KB boundary, accesses the TLB in a one-cycle window and hits the entry which maps its virtual address, VA1, to physical address PA1.
3. The load triggers a prefetch request based on PA1 which might be outside of the page boundary for PA1, but within the 64KB aligned physical address region containing PA1.

Implications

If the above conditions are met, the core might generate an unexpected read to a physical address within the 64KB aligned physical address region of the load.

Workaround

Arm does not expect read-sensitive Device memory to be mapped to a physical address which overlaps with a 64KB aligned physical address region belonging to Normal memory, therefore no workaround is necessary.

1619801
TRCIDR3.CCITMIN value is incorrect

Status

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

Description

Software reads of the TRCIDR3.CCITMIN field, corresponding to the instruction trace counting minimum threshold, observe the value 0x100 or a minimum cycle count threshold of 256. The correct value should be 0x4 for a minimum cycle count threshold of 4.

Configurations Affected

This erratum affects all configurations.

Conditions

- Software reads the TRCIDR3 ID register.
- Software uses the value of the CCITMIN field to determine minimum instruction trace cycle counting threshold to program the ETM.

Implications

If software uses the value returned by the TRCIDR3.CCITMIN field, then it will limit the range which could be used for programming the ETM. In reality, the ETM could be programmed with a much smaller value than what is indicated by the TRCIDR3.CCITMIN field and function correctly.

Workaround

The value for the TRCIDR3.CCITMIN field should be treated as 0x4.

1619805
ESB instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE

Status

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

Description

If a Virtual SError is pending and masked at the current Exception level when an Error Synchronization Barrier (ESB) instruction is executed, then the VDISR_EL2 update occurs properly, but sometimes the clearing of HCR_EL2.VSE might not occur. This failure to clear HCR_EL2.VSE can only occur when the Virtual SError is masked.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. A Virtual SError is pending at the current Exception level.
2. Virtual SErrors are masked at the current Exception level.
3. An ESB instruction executes.

Implications

If the above conditions are met, then under specific microarchitectural timing conditions HCR_EL2.VSE might not be cleared to 0, which the Arm architecture requires. This might result in spurious Virtual SErrors. Under all circumstances, the Virtual SError syndrome from VSESR_EL2 is correctly recorded in VDISR_EL2, and VDISR_EL2.A is correctly set to 1.

Workaround

A workaround should not be required because existing software only executes ESB instructions at EL2 and above. If your software executes ESB instructions at EL1 with the conditions that are described above, then contact Arm support for more details.

1619808
The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error

Status

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

Description

When an address matching breakpoint is set to the instruction following an instruction that is affected by a parity error, the core might detect a breakpoint exception on the instruction with the parity error.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in AArch64 state.
2. An instruction that is cached in L0 Macro-op cache has a parity error.
3. An address matching breakpoint is marked on the instruction right after the above parity error instruction.

Implications

If the above conditions are met, then the core might detect a breakpoint exception at the instruction with the parity error, which is incorrect.

Workaround

This erratum has no workaround.

1619809
PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled

Status

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

Description

Virtual Issue Queue size reduction for dynamic power savings fails to disengage when PDP is disabled.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core enables Performance defined power (PDP) optimization.
2. PDP engages power savings, due to low utilization of parts of issue queues.
3. While engaged PDP is disabled, either by pin control or by system register access to CPUPPMCR_EL3

Implications

If the above conditions are met, the parts of the issue queue that are turned off for power will not be turned on when PDP is disabled. This results in lower performance and lower power. The regain of performance is expected, even at the cost of higher power, when PDP is disabled.

Workaround

There are 2 options to workaround this issue.

1. Never enable PDP feature by ensuring that boot code never sets CPUPPMCR_EL3[49:48] to a non-zero value.
2. Once PDP feature is enabled, never disable it. If use of PDP feature is desired, boot code must set CPUPPMCR_EL3[49:48] to 2'b10 and never change.

1619811
The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception

Status

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

Description

When a T32 instruction is affected by parity error and the next instruction is marked as an address matching breakpoint, the core might deadlock or detect an address matching breakpoint at an incorrect location.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core fetches a T32 instruction from the L1 instruction cache.
2. Either L1 instruction cache tag RAM or L1 instruction cache data RAM has a parity error on an entry associated with the T32 instruction.
3. An address matching breakpoint exception is programmed on the next instruction after the T32 instruction.

Implications

If the above conditions are met, then the core might behave in one of the following ways:

1. The core might stall until an asynchronous exception, such as a timer interrupt, occurs on the core.
2. The core might detect a breakpoint exception at the instruction affected by the parity error, which is incorrect.

Workaround

This erratum has no workaround.

1619813
Enabling L2 cache partitioning might result in a loss of performance

Status

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

Description

When the L2 cache is configured to be partitioned between lines containing data and instruction, it might restrict the number of ways that can be allocated.

Configurations Affected

The erratum affects all configurations.

Conditions

L2 cache partitioning is enabled by setting either CPUECTLR_EL1[60:58] or CPUECTLR_EL1[57:55] to a non-zero value.

Implications

Setting CPUECTLR_EL1[60:58] to a non-zero-value restricts the number of ways that can be allocated by lines containing instruction. Similarly, setting CPUECTLR_EL1[57:55] to a non-zero value restricts the number of ways that can be allocated by lines containing data. This might have an impact on performance.

Workaround

This erratum can be avoided by disabling L2 cache partitioning by setting either CPUECTLR_EL1[60:58] or CPUECTLR_EL1[57:55] to zero.

1619815
ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort

Status

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

Description

ECC error or external data abort, seen by a speculative instruction in the shadow of a flush caused by a precise exception due to another non-speculative instruction, can lead to ESR and FAR registers being corrupted.

Configurations Affected

The erratum affects all configurations.

Conditions

1. Precise exception due to a non-speculative instruction results in a flush.
2. A speculative instruction encounters an ECC error or external data abort in the shadow of the flush.
3. The speculative instruction reports this ECC error or external data abort as a precise exception.

Implications

If the above conditions are met, ESR and FAR registers could be corrupted.

Workaround

There is no workaround for this erratum.

1619816
A load to normal memory might trigger a prefetch request outside of the current mapped page

Status

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

Description

A load to a page mapped as Normal Write-Back memory using a 4KB or 16KB page size might result in a prefetch request to a physical address that resides outside of the current mapped page, but within the aligned 64KB region.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The system has mapped read-sensitive Device memory or Normal Non-Cacheable and Normal Write-Back memory using 4KB or 16KB pages within the same aligned 64KB region.
2. A load to Normal Write-Back memory might trigger a hardware prefetch to a physical address outside the 4KB or 16KB page, but within the aligned 64KB region, targeting a region mapped as Device memory.

Implications

If the above conditions are met, then the Processing Element (PE) might generate a speculative read to read-sensitive device or
generate a speculative read to Normal Non-Cacheable memory and cache its content.

Workaround

Arm does not expect Device memory and Normal memory to be mapped within the same 64KB memory region. Normal Non-Cacheable and Normal Write-Back can be within the same 64KB memory region. There is no workaround for the latter.

1619817
RAS error status records could log spurious corrected error

Status

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

Description

Under certain conditions, a single spurious corrected error from the L1 Data RAM might be logged into the RAS error status registers following a real corrected error that has been logged.

Configurations Affected

The erratum affects all configurations.

Conditions

1. Single-bit error (SBE) is detected in the L1 Data RAM.
2. There are back to back capacity evictions on the L1 Data RAM, the first of which is to the cache line with SBE detected.

Implications

If the above conditions are met, then:

1. ERR0MISC0_EL1.CECO might be incremented for both the real SBE and a spurious error.
2. ERR0MISC0_EL1.OFO might be set to indicate the corrected error count, other, has overflowed.
3. ERR0MISC0_EL1.CECR might be incremented for both the real SBE and a spurious error.
4. ERR0MISC0_EL1.OFR might be set to indicate the corrected error count, repeat, has overflowed.
5. nFAULTIRQ[0] might be set by the spurious corrected error if either counter has overflowed and ERR0CTLR_EL1.CFI is set.

Workaround

There is no workaround for this erratum. The effects can be mitigated by allowing for a larger number of corrected errors to cause overflow conditions in the ERR0MISC0_EL1.CECO and ERR0MISC0_EL1.CECR fields.

1629078
ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly

Status

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

Description

When an ECC error is detected to an entry in the L2 data ram that does not contain valid data, the RAS reporting that is associated with that fault does not match expectation.

Configurations Affected

All configurations are affected.

Conditions

A single or double bit ECC error occurs during a read of an L2 data ram entry not containing valid data.

Implications

If the conditions occur the error is typically not reported.

If the conditions occur and the error is reported, ERR0STATUS.AV will be incorrectly set to 1. Other fields of the error record will be correct.

Workaround

No workaround is required.

1651942
MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption

Status

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

Description

In AArch32, MRC reads of DBGDSCRint into destination APSR_nzcv (Rt=15) always produce a result of 0. Also, if there is a younger MRC or MRRC read to any accessible register following the DBGDSCRint read into APSR_nzcv, then the younger read result might be corrupted.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in AArch32 state at EL0.
2. An MRC read of DBGDSCRint into APSR_nzcv (Rt=15) occurs.

Implications

If the above conditions are met, then:

1. APSR_nzcv is always written with 0.
2. Under specific microarchitectural timing conditions in AArch32 EL0, a subsequent MRC or MRRC might be corrupted.

Workaround

Directly read DBGDSCRint with an MRC instruction into a general-purpose register (R0-R14), and then write that general-purpose register to the flags by doing an MSR APSR_f. To avoid the possible corruption, add an ISB instruction before any subsequent MRC or MRRC instructions.

1656369
APB access to trace registers does not work during Warm reset

Status

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

Description

During Warm reset, APB writes to trace registers are ignored, and reads return incorrect data. Trace continues through Warm reset over the ATB interface as expected.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Warm reset is asserted.
2. Trace registers are accessed over the APB interface.

Implications

If the above conditions are met, then APB writes to the trace registers are ignored. APB reads to the trace registers return incorrect data.

Workaround

The workaround for this erratum is to set up the trace registers in the needed configuration before entering Warm reset.

1657962
Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock

Status

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

Description

Under certain conditions, executing a cache maintenance by set/way instruction targeting the L1 data cache in close proximity to multiple snoops where the older snoop detects a transient ECC error might result in a deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core has executed at least two snoop requests looking up the L1 data cache. These could have been generated internally from this core or from another core in the system.
2. The older snoop detects a transient single-bit or double-bit ECC error, but at least two snoops have performed a lookup of the L1 data cache.
3. The core executes a cache maintenance by set/way instruction targeting the L1 data cache.
4. The snoops are required to perform another lookup due to the ECC error detected. All snoops are rescheduled to maintain ordering of the snoop transactions.
5. The snoop transactions continuously retry the L1 data cache lookup, preventing the cache maintenance operation from completing.

Implications

If the above conditions are met under certain timing conditions, then the snoops might not make progress, resulting in a deadlock. Arm does not expect cache maintenance operations by set/way to be executed in most code sequences, since hardware mechanisms have been incorporated for flushing the caches as a part of powerdown sequences. Software is expected to use cache maintenance operations by VA to manage coherency.

Note that cache maintenance by set/way instructions are UNDEFINED at EL0.

Workaround

Software should avoid the use of cache maintenance operations by set/way. A hypervisor should trap these instructions by setting HCR_EL2.TSW = 1 and emulate the instructions with equivalent cache maintenance operations by virtual address for the entire address space of the guest.

1659168
A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERR0MISC0_EL1 and ER0ADDR_EL1

Status

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

Description

Under certain conditions, a load that observes a double-bit ECC error (DBE) in the L1 data cache data RAM after a snoop observed a single-bit ECC error (SBE) might result in incorrect information being recorded in the ERR0MISC0_EL1 and ER0ADDR_EL1 registers.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A snoop detects an SBE in the L1 data cache tag RAM.
2. A load detects a DBE error in a particular set and way of the L1 data cache data RAM around the same time as the snoop detected the SBE.
3. The ECC detected by the snoop and load are to the same way but not necessarily to the same set.

Implications

If the above conditions are met, then ERR0MISC0_EL1.SUBARRAY, ERR0MISC0_EL1.WAY, and ER0ADDR_EL1 might have incorrect values. The remaining fields of the ERR0MISC0_EL1 register remain correct.

Workaround

There is no workaround for this erratum.

1662728
Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior

Status

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

Description

A PE that performs a data cache clean and invalidate instruction (DC CIVAC) to memory locations that contain code which is being actively modified by another PE might not execute the newly written code, which is required by Arm architecture.

Configurations Affected

This erratum affects configurations with instruction cache coherency enabled.

Conditions

1. PE0 writes new instructions to memory location A, and sets a flag indicating that the new code is ready:
   a. Store A, <new code>
   b. DMB
   c. store flag B
2. PE1 executes a DC CIVAC instruction to location A
3. PE1 executes:
   a. LOOP: load flag B
   b. CBZ LOOP
   c. ISB
   d. branch A

Implications

If the above conditions are met, PE1 might not execute the code that is written by PE0, which required by the Arm architecture. Arm does not expect that the code sequence that is executed by PE1 appears in normal code.

Workaround

If the DC CIVAC executed by PE1 is necessary, follow it with a DMB instruction.

1690750
The core might record incorrect INDEX into ERR0MISC0 when L0 Macro-op cache is affected by parity error

Status

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

Description

The CPU might update ERR0MISC0 register incorrectly when L0 Macro-op cache is affected by parity error.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A core detects a parity error in the L0 Macro-op cache with certain timing.

Implications

ERR0MISC0[18:6] might record RAM index which was not affected by the parity error. All other fields track correct information.

ERR0MISC0.INDEX should not be used as part of Functional Safety processing of faults from the L0 Macro-op cache. No extraneous faults or errors will be reported. No faults or errors will be missed.

Workaround

This erratum has no workaround.

1694300
Instruction sampling bias exists in SPE implementation

Status

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

Description

A PE that is used to perform instruction sampling using the SPE mechanism might exhibit sampling bias toward instructions that are branch targets.

Configurations Affected

This erratum affects all configurations.

Conditions

1. SPE configured and utilized on PE.

Implications

Software utilizing SPE might see unexpectedly high sample counts for branch target instructions and unexpectedly low sample counts for some instructions closely following a branch target.

Workaround

No hardware workaround.

1702491
The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB

Status

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

Description

The CPU might not update IDATA*_EL3 correctly when a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB is initiated.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A direct memory access to L1 Instruction Cache Tag is initiated while the core is processing IC IALLU or IC IALLUIS.
2. A direct memory access to L1 Instruction TLB is initiated while an address translation was disabled in EL3.

Implications

If one of the above conditions are met, IDATA*_EL3 might not be updated after the completion of the direct memory access. IDATA*_EL3 might hold an old value for L1 Instruction Cache Tag access or a corrupted value for L1 Instruction TLB access.

Workaround

This erratum has no workaround.

1702951
Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice

Status

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

Description

Execution of load instructions from the Instruction Transfer Register in Debug state might result in the instruction being executed twice before returning control to the debugger.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in Debug state.
2. A load instruction is written to the External Debug Instruction Transfer Register (EDITR) via the external debug interface.
3. Certain internal timing conditions relating to execution of a previous load instruction exist.

Implications

If the above conditions are met, then the instruction might execute twice before returning control to the debugger. If the instruction executes twice and the load is from Device memory, then corruption of memory read pointers might result. If the instruction executes twice and base register writeback is involved, then the second load will be from a different address (corrupting the load destination register), and the base address register will be corrupted.

Workaround

A workaround is only needed if there is any possibility of connecting an external debugger to the core. If that possibility exists, setting CPUACTLR3_EL1[47] in the boot sequence will prevent this behavior. There is no performance impact associated with setting this bit, but there is a potential (workload dependent) power increase of approximately 1.5% total core power.

1740839
RAS error reported could have incorrect value in ERR0ADDR_EL1

Status

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

Description

Under certain conditions, capacity eviction of a line which single or double bit ECC error is in the process of being reported could end up corrupting the value in ERR0ADDR_EL1 register.

Configurations Affected

The erratum affects all configurations.

Conditions

1. ECC error is detected in the L1 Data RAM.
2. RAS error is in the process of being reported and the line is replaced due to capacity eviction.

Implications

If the above conditions are met, ERR0ADDR_EL1 could have incorrect value. In the case of a single bit error, the data will be corrected and in the case of a double bit error data is written out as poisoned.

Workaround

There is no workaround for this erratum.

1755651
MPAM value associated with descriptor fetch requests could be incorrect

Status

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

Description

The MPAM value associated with descriptor fetch requests when translating PLDW or cache maintenance instructions at EL0 could be incorrect.

Configurations affected

This erratum affects all configurations.

Conditions

1. Processing Element (PE) executes PLDW or cache maintenance instruction at EL0.
2. A table walk request is initiated on behalf of the instruction.

Implications

If the above conditions are met, then the PE generates a memory-system request for translation table walk using MPAM information for EL2 if HCR_EL2.{E2H, TGE} == {1,1} or EL1 otherwise.

Workaround

There is no workaround.

1771937
Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR

Status

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

Description

If the watchpoint address targets a lower portion of a cache line, but not all of the cache line, and the address target of the Data Cache Zero by VA (DC ZVA) falls in the upper portion of the cache line that the watchpoint does not target, the Fault Address Register (FAR) (or External Debug Watchpoint Address Register (EDWAR) if setup for Debug Halt) will contain an incorrect address.

Configurations affected

This erratum affects all configurations.

Conditions

1. Watchpoint targets double word (or less or more) at address A.
2. DC ZVA targets address greater than A+7, but less than A+63. The cache line size is 64 bytes, which is a mis-aligned address.

Implications:

FAR contains target address of DC ZVA.
EDWAR contains target address of DC ZVA if enabled for Debug Halt.

Workaround:

There is no hardware workaround. The common case for DC ZVA targets is to be granule aligned, thus most software will not be affected by this case.

1776362
A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data

Status

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

Description:

A memory mapped write to PMSSRR at offset 0x6f4 might configure the Cycle counter and/or Performance Monitor event counters to be reset along with reset of corresponding overflow status bits in the PMOVSR register. The register supports read/write functionality instead of RAZ/WI.

Configurations affected

This erratum affects all configurations.

Conditions

1. System enables PMU snapshot mechanism.
2. System performs memory mapped write of PMSSRR setting PMSSRR[x], where x is 31 or any value from 0 to 5 (inclusive).
3. Snapshot trigger is seen through a legal mechanism.

Implications

If the above conditions are met, the corresponding counter (PMCCNTR_EL0 if x=31 or PMEVCNTR<x>_EL0 if x = [0,5]) will reset after a snapshot is taken. Further, the corresponding bit in the PMOVSR_EL0 register will be reset.

A memory mapped read will return data that is written to these bits and 0 otherwise.

This register is supposed to have RAZ/WI functionality and no effect on other counters.

Workaround

Avoid write of PMSSRR when system is using the PMU Snapshot mechanism.

1778872
Possible loss of CTI event

Status

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

Description

A CTI event from the core to the external DebugBlock might be dropped, in rare occurrences, if close in temporal proximity to a previous CTI event.

Configurations affected

This erratum affects all configurations.

Conditions

1. CTI event occurs.
2. Another CTI event occurs before completion of the processing of the previous CTI event.

Implications

CTI events might be dropped.

Workaround

This erratum has no workaround.

1784478
Loss of CTI events during warm reset

Status

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

Description

ETM external output CTI events from the core to the external DebugBlock will not be reported during warm reset.

Configurations affected

This erratum affects all configurations.

Conditions

1. An ETM external output CTI event occurs while warm reset is asserted.

Implications

The ETM external output CTI event will be dropped and any cross triggering that depends on this CTI event will not occur. For example, if the ETM external output was to be used to trigger a trace capture component to stop trace capture, then trace capture will not stop due to this event.

Workaround

This erratum has no workaround.

1791399
The core might deadlock when an external debugger injects instructions using ITR register

Status

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

Description

The core might deadlock when an external debugger injects instructions by ITR register.

Configurations affected

This erratum affects all configurations.

Conditions

1. An external debugger requests the core to enter debug state while the core is stalled because of an instruction abort due to a permission fault.
2. The external debugger injects instructions using the ITR register.

Implications

The core might deadlock if the above conditions are satisfied.

Workaround

This erratum has no workaround.

1794808
Uncorrectable tag errors in L2 cache might cause deadlock

Status

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

Description

Under rare conditions that include the aliasing of multiple virtual addresses to a single physical address, a detected and reported double-bit ECC error in the L2 cache tag RAM might lead to a state in which an unexpected L1 cache eviction can cause a deadlock in the L2 cache.

Configurations affected

This erratum affects all configurations.

Conditions

1. L2 cache detects and reports a tag double-bit ECC error.
2. A set of rare conditions occur within the PE's memory system.

Implications

If the above conditions are met, the L2 transaction queue might deadlock and never complete the prefetch operation.

Workaround

There is no workaround for this erratum.

1803672
Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents

Status

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

Description

Memory uploads via memory access mode within Debug state might fail to set EDSCR.TXfull to 1, possibly resulting in an intended memory read being skipped and erroneous memory contents being displayed for that address.

Memory downloads via memory access mode within Debug state might prematurely clear EDSCR.RXfull, possibly resulting in an intended memory write being skipped and subsequent memory access mode downloads therefore writing data to incorrect addresses.

Configurations affected

This erratum affects all configurations.

Conditions

For memory upload:

1. The core is in Debug state having been properly set up via the external debug interface for memory upload (target to external host).
2. A series of external reads from DBGDTRTX_EL0 are used, where each read first clears EDSCR.TXfull to 0, then initiates memory uploads via PE-generated load & system register write instruction pairs, then sets EDSCR.(TXfull,ITE) to (1,1) on successful completion of each iteration.
3. Certain internal timing conditions relating to execution of a previous load instruction exist, resulting in the failure to set EDSCR.TXfull to 1 on some iteration.

For memory download:

1. The core is in Debug state having been properly set up via the external debug interface for memory download (external host to target).
2. A series of external writes to DBGDTRRX_EL0 are used, where each write first sets EDSCR.RXfull to 1, then initiates memory downloads via PE-generated system register read & store instruction pairs, then sets EDSCR.(RXfull,ITE) to (0,1) on successful completion of each iteration.
3. Certain internal timing conditions relating to execution of a previous load instruction exist, resulting in a premature clearing of EDSCR.RXfull to 0 on some iteration.

Implications

If the above conditions are met, the failure mechanism could effectively skip an intended memory read in a memory upload loop, thus resulting in the erroneous display of data associated with the affected memory address. Or, the failure mechanism could effectively skip an intended memory write in a memory download loop, thus resulting in subsequent memory access mode downloads writing data to incorrect addresses.

Workaround

A workaround is only needed if there is any possibility of connecting an external debugger to the core. If that possibility exists, then there are 2 separate workarounds:

1. Perform the memory upload or download operations with the debugger's FAST_MEMORY_ACCESS disabled. This can impact the performance of memory upload and download operations in Debug state, resulting in slight visible delays in the debugger user interface on memory upload and longer download times.
   or
2. Set CPUACTLR3_EL1[47] in the boot sequence to prevent the faulty behavior. There is no performance impact associated with setting this bit, but there is a potential (workload dependent) power increase of approximately 1.5% total core power.

1817602
Persistent faults on speculative elements of SVE First-fault gather-load instructions might result in deadlock

Status

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

Description

Double-bit ECC errors or bus faults on a non-first active element of an SVE First-fault gather load might result in deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

1. A double-bit ECC error or bus fault occurs on a non-first active element of an SVE First-fault gather-load instruction.
2. The double-bit ECC error or bus fault must be persistent, meaning on repeated attempted executions of the instruction the fault consistently re-occurs.
3. Certain internal timing conditions concerning the execution order of the non-first active element relative to the first active element exist.

Implications

If the above conditions are met, the core can deadlock. An interrupt can be recognized, but the SVE First-faulting gather-load instruction makes no forward progress in the presence of persistent ECC double-bit errors or bus faults that occur on elements other than the first active element.

Workaround

No workaround is expected to be required. RAS error handling and recovery techniques cover this scenario.

1825496
External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior

Status

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

Description

In Debug state with SCTLR_ELx.IESB set to 1, memory uploads and downloads executed in memory access mode might lead to unpredictable behavior for the current exception level.

Configurations affected

This erratum affects all configurations.

Conditions

1. Core is In Debug state.
2. SCTLR_ELx.IESB is set to 1 for the current exception level.
3. Memory access mode is enabled via EDSCR.MA set to 1.

Implications

If the above conditions are met, memory upload and download behavior is unpredictable for the current exception level and might lead to incorrect operation or results. The unpredictable behavior is limited to legal behavior at the current exception level.

Workaround

The erratum can be avoided by clearing SCTLR_ELx.IESB before performing memory uploads or downloads in Debug state using memory access mode.

1825527
Transient L2 tag double bit Errors might cause data corruption

Status

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

Description

Under certain uncommon conditions, transient double bit tag errors might cause valid cache data that is in an unrelated line in the same set to be overwritten.

Configurations affected

This erratum affects all configurations.

Conditions

The following conditions must be met during additional rare timing and state conditions:

1. A double bit error (DBE) in the tag occurs shortly after the read of a line.
2. The DBE occurs before a write to that same line in a different way.
3. The DBE corrects after the write to that line.
4. An additional read is made to that line before it is evicted from the cache.

Implications

If the above conditions are met, the data in an unrelated line in the same set might be overwritten and corrupted. The effect on the failure rate is negligible in such a case. There is still substantial benefit being gained from the ECC logic.

Workaround

There is no workaround.

1835961
MPAM value associated with MMU descriptor fetch requests might be incorrect

Status

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

Description

The MPAM value associated with MMU descriptor fetch requests could be incorrect when performing address translation on behalf of *Address Translation* (AT) instructions, sampled instructions for statistical profiling, the *Statistical Profiling Extension* (SPE) buffer, or the table walk prefetcher.

Configurations Affected

This erratum affects all configurations.

Conditions

The MPAM value could be incorrect when:

1. An MMU descriptor fetch request is made on behalf of AT instructions, statistically profiled sampled instructions, the SPE buffer, or the table walk prefetcher.

Implications

If the above condition is met, the core generates a memory-system request for the translation table walk using the MPAM ID for the:

1. Context associated with a AT instruction instead of the context that executed the instruction.
2. EL1 or EL2 context when HCR_EL2 {E2H, TGE} == {1,1} instead of EL0 when a sampled instruction for statistical profiling is executed in user mode.
3. Current context instead of SPE owning translation regime for buffer translation requests.
4. EL0 context for translations generated by table walk prefetcher trained in EL1 or EL2 context when HCR_EL2{E2H, TGE} == {1,1}.

Workaround

There is no workaround.

1846398 ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect

Status

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

Description

Under certain conditions, the ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values recorded for ECC errors in the L1 data cache might be incorrect.

Configurations affected

This erratum affects all configurations.

Conditions

1. The L1 data cache contains both a single-bit and double-bit ECC error on different words within the same 64-byte cacheline.
2. An access is made to the cacheline in the L1 data cache containing both the single-bit and double-bit ECC errors simultaneously.

Implications

If the above conditions are met, then ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE might have an incorrect values.

Workaround

There is no workaround for this erratum.

1853757
The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified

Status

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

Description

When a core fetches an instruction from a virtual address that is associated with a page table entry which has been modified and the fetched block is affected by parity error, the core might report an incorrect address within the same 32B block onto the Fault Address Register (FAR).

Configurations Affected

All configurations are affected.

Conditions

1. The core fetches instructions from an aligned 32B virtual address block.
2. A page table entry associated with the above 32B aligned block is updated. The new translation would cause an instruction abort.
3. TLB holds the old translation since the synchronization process, for example, TLB Invalidate (TLBI) followed by Data Synchronization Barrier (DSB), was not completed.
4. Some of the fetched instructions are affected by parity error in I-cache data RAM.
5. Context synchronization events were not processed between the last executed instruction and the above instruction.

Implications

When the above conditions are satisfied, a core might report an incorrect fetch address to FAR_ELx. The address reported in FAR_ELx points at an earlier location in the same aligned 32B block. FAR_ELx[63:5] still points correct virtual address.

Workaround

There is no workaround.

1858170
Incorrect value reported for SPE PMU event SAMPLE_FEED

Status

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

Description

Under certain conditions when a CMP instruction is followed by a Branch, the SAMPLE_FEED PMU event 0x4001 is not reported.

Configurations Affected

This erratum affects all configurations.

Conditions

1. *Statistical Profiling Extension* (SPE) sampling is enabled
2. SPE samples a CMP instruction, which is followed immediately by a BR instruction.

Implications

If the above conditions are met, then the SAMPLE_FEED event may not be incremented.

For most expected use cases, the inaccuracy is not expected to be significant.

Workaround

There is no workaround.

1872062
L2 data RAM may fail to report corrected ECC errors

Status

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

Description

For specific operation types and cache states, a read of the L2 data RAM might fail to report a detected and corrected single-bit ECC error.

Configurations Affected

All configurations are affected.

Conditions

1. PE L1 data cache and L2 cache are in a SharedClean state and the exclusive monitor is armed for a given physical address.
2. PE executes a store exclusive instruction to this physical address.
3. L2 cache reads its data RAMs, and detects and corrects a single-bit ECC error.

Implications

If the above conditions are met, the PE will correct the error, but might fail to report it in the RAS error log registers. This can cause a small loss in diagnostic capability.

Workaround

There is no workaround.

1878813
PFG duplicate reported faults through a Warm reset

Status

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

Description

Under certain conditions, the Pseudo-fault Generation Error Record Registers might generate duplicate faults through a Warm reset.

Configurations Affected

All configurations are affected.

Conditions

1. ERR0PFGCDN is set with a non-zero countdown value.
2. ERR0PFGCTL is set to generate a pseudo-fault with ERR0PFGCTL.CDEN enabled.
3. The countdown value expires, generating a pseudo-fault.
4. Warm reset asserts.

Implications

After the Warm reset, a second generated pseudo-fault might occur.

Workaround

De-assert the ERR0PFGCTL control bits before asserting a Warm reset.

1880117
Noncompliance with prioritization of Exception Catch debug events

Status

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

Description

ARMv8.2 architecture requires that Debug state entry due to an Exception Catch debug event (generated on exception entry) occur before any asynchronous exception is taken at the first instruction in the exception handler. An asynchronous exception might be taken as a higher priority exception than Exception Catch and the Exception Catch might be missed altogether.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Debug Halting is allowed.
2. EDECCR bits are configured to catch exception entry to ELx.
3. A first exception is taken resulting in entry to ELx.
4. A second, asynchronous exception becomes visible at the same time as exception entry to ELx.
5. The second, asynchronous exception targets an Exception level ELy that is higher than ELx.

Implications

If the above conditions are met, the core might recognize the second exception and not enter Debug state as a result of Exception Catch on the first exception. When the handler for the second exception completes, software might return to execute the first exception handler, and assuming the core does not halt for any other reason, the first exception handler will be executed and entry to Debug state via Exception Catch will not occur.

Workaround

When setting Exception Catch on exceptions taken to an Exception level ELx, the debugger should do either or both of the following:

1. Ensure that Exception Catch is also set for exceptions taken to all higher Exception Levels, so that the second (asynchronous) exception generates an Exception Catch debug event.
2. Set Exception Catch for an Exception Return to ELx, so that when the second (asynchronous)

exception handler completes, the exception return to ELx generates an Exception Catch debug event.

Additionally, when a debugger detects that the core has halted on an Exception Catch to an Exception level ELy, where y > x, it should check the ELR_ELy and SPSR_ELy values to determine whether the exception was taken on an ELx exception vector address, meaning an Exception Catch on entry to ELx has been missed.

1890031
Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO

Status

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

Description

If a Corrected Error is recorded because of a bus error which has no valid location (ERR0STATUS.MV=0x0), then a subsequent Corrected Error might incorrectly increment either of the ERR0MISC0.CECR or ERR0MISC0.CECO counters.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A Corrected Error which has no valid location (ERR0STATUS.MV=0x0) is recorded.
2. A subsequent Corrected Error occurs.

Implications

The subsequent Corrected Error might improperly increment either of the ERR0MISC0.CECR or ERR0MISC0.CECO counters.

Workaround

No workaround is expected to be required.

1923196
IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB

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

Description

After implementation-defined RAMINDEX register is programmed to initiate direct memory access to internal memory for Instruction TLB, implementation-defined IDATAn_EL3 value represents unpredictable value.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Implementation-defined RAMINDEX register is programmed to initiate direct memory access to internal memory for Instruction TLB.

Implications

If the above conditions are met, IDATAn_EL3 register might represent incorrect value for Translation regime, VMID, ASID, and VA[48:21].

Workaround

There is no workaround.

1925786
Unsupported atomic fault due to memory type defined in first stage of translation might result in exception being taken to EL2

Status

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

Description

Under certain conditions, when far atomics are not supported by the system, an unsupported atomic fault due to the memory type defined in the first stage of translation might result in an exception being taken to EL2.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. Device memory type is defined in the first stage of translation for the atomic instruction.
2. HCR_EL2.VM, HCR_EL2.FWB, and HCR_EL2.CD bits are set.
3. The system does not support far atomics.

Implications

If the above conditions are met, an exception is incorrectly taken to EL2.

Workaround

There is no workaround.

1937142
MPAM value associated with translation table walk request might be incorrect

Status

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

Description

Under some scenarios, MPAM value associated with translation table walk request might be incorrect when context changes along with security state.

Configurations Affected

All configurations are affected.

Conditions

1. Translation table walk request attempted before a context switch but is not completed until after a context change where the secure state changes.

Implications

MPAM value associated with the table walk request might be incorrect.

Workaround

There is no workaround.

1949546
The PE might deadlock if Pseudofault Injection is enabled in Debug State

Status

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

Description

If Pseudofault Injection is enabled for the PE node (ERR0PFGCTL.CDNEN=0x1) and the PE subsequently enters Debug State, then the PE might deadlock. Alternatively, if the PE is executing in Debug State and the PE enables Pseudofault Injection for the PE node (ERR0PFGCTL.CDNEN=0x1), then the PE might deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. ERR0PFGCTL.CDNEN is set to 0x1 to enable Pseudofault Injection.
2. The PE enters Debug State.

OR

1. The PE is executing in Debug State.
2. ERR0PFGCTL.CDNEN is set to 0x1 to enable Pseudofault Injection.

Implications

If the above conditions are met, then the PE might deadlock.

Workaround

Ensure ERR0PFGCTL.CDNEN=0x0 before entering Debug State and while executing in Debug State.

1960366
Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC

Status

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

Description

A statistical profiling buffer translation request which encounters multiple hits in the TLB might report an incorrect fault status code in PMBSR_EL1.FSC.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Statistical Profiling Extension (SPE) is enabled.
2. A translation request is made for the statistical profiling buffer.
3. This translation request encounters multiple hits in the TLB due to incorrect invalidation or misprogramming of translation table entries.

Implications

If the above conditions are met, then the fault status code reported in PMBSR_EL1.FSC might incorrectly indicate an illegal or incorrect fault status code instead of the correct TLB Conflict fault code.

Workaround

There is no workaround.

1965494
Incorrect timestamp value reported in SPE records when timestamp capture is enabled

Status

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

Description

The timestamp value that is captured in SPE records is from when the SPE record is written out to L2, as opposed to before the operation is signaled as "complete".

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

If the above conditions are met, then the timestamp value reported in the SPE records might be outside of the sampled operation's lifetime.

For most expected use cases, the inaccuracy is not expected to be significant.

Workaround

There is no workaround.

1986030
Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes

Status

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

Description

Collision information captured by PMBSR_EL1.COLL might be lost under certain circumstances, when the buffer management interrupt is raised.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A sampling collision event is detected.
2. Subsequent SPE write results in 2 SEI errors.

Implications

If the above conditions are met, the collision indicator in PMBSR_EL1 is incorrectly set to 0, following the 2nd SEI error. PMBSR_EL1 does capture and set the "Data Loss" (DL) indicator and all the other PMBSR_EL1 fields correctly.

Workaround

There is no workaround for this erratum.

1987045
AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFI or WFE state

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround.

2001721
DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL

Status

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

Description

In Debug state with SCTLR_ELx.IESB set to 1, the **DRPS** (debug only) instruction does not execute properly. Only partial functionality of the **DRPS** instruction is performed.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core is in Debug state.
2. SCTLR_ELx.IESB is set to 1 for the current exception level.
3. The **DRPS** instruction is executed.

Implications

If the above conditions are met, the **DRPS** instruction does not complete as intended, which might lead to incorrect operation or results. Register data or memory will not be corrupted. There are also no security or privilege violations.

Workaround

The erratum can be avoided by clearing SCTLR_ELx.IESB followed by the insertion of an **ISB** and an **ESB** instruction in code before the **DRPS** instruction.

2020786
CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation

Status

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

Description

When an instruction fetch is initiated for a page programmed as non-cacheable normal memory in stage-1 translation and as device memory in stage-2 translation, the instruction memory might incorrectly return 0. This might cause an unexpected UNDEFINED exception.

Configurations Affected

The erratum affects configurations with COHERENT_ICACHE=0.

Conditions

This erratum occurs under the following conditions:

1. A CPU fetch instruction from a page satisfies the following:
    - Stage-1 translation of this page is programmed as non-cacheable normal memory.
    - Stage-2 translation of this page is programmed as device memory.

Implications

If the above conditions are met, the CPU might read 0 from the instruction memory. This instruction might cause an unexpected UNDEFINED exception. Instruction fetches to device memory are not architecturally predictable in any case, and device memory is expected to be marked as execute never, so this erratum is not expected to cause any problems to real-world software.

Workaround

This erratum has no workaround.

2033528
ETM trace information records a branch to the next instruction as an N atom

Status

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

Description

If a branch is taken to the next instruction, and if the instruction state remains the same, then the ETM traces it as an N atom rather than an E atom or branch address packet. This is incorrect as the ETM architecture says a taken branch should be traced as an E atom. This affects all forms of branches. State-changing branches are traced correctly.

Configurations Affected

This erratum affects all configurations.

Conditions

This issue might occur when:

1. ETM is enabled.
2. A branch is taken to the next instruction.
3. The instruction state does not change.

Implications

A trace decoder that interprets an N atom to move to the next instruction in the same state without a push or pop from the return stack will correctly maintain the control flow but will not be able to infer anything from a conditional branch.

A trace decoder that checks if unconditional branches were not traced as N atom might report an error.

Workaround

To ensure continued control flow, ensure the trace decoder always interprets an N atom to move to the next instruction in same state without a push or pop from the return stack.

2052426
An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers

Status

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

Description

When an **MSR** instruction and an APB write operation are processed on the same cycle, the **MSR** instruction might not update the destination register correctly.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. A CPU executes an **MSR** instruction to update any of following SPR registers:
   a. DBGBCR<n>_EL1
   b. DBGBVR<n>_EL1
   c. DBGWCR<n>_EL1
   d. DBGWVR<n>_EL1
   e. OSECCR_EL1
2. An external debugger initiates an APB write operation for any of following registers:
   a. DBGBCR<n>
   b. DBGBVR<n>
   c. DBGBXVR<n>
   d. DBGWCR<n>
   e. DBGWVR<n>
   f. DBGWXVR<n>
   g. EDECCR
   h. EDITR
3. The SPR registers (for example, OSLSR_EL1.OSLK and EDSCR.TDA) and external pins are programmed to allow the following behavior:
   a. The execution of an **MSR** instruction in condition 1 to update its destination register without neither a system trap nor a debug halt
   b. The APB write operation in condition 2 to update its destination register without error
4. The **MSR** instruction execution in condition 1 and APB write operation in condition 2 happen in same

cycle.
5. The **MSR** write and the APB write are to two different registers. The architecture specifies that it is the software or debugger's responsibility to ensure writes to the same register are updated as expected.

Implications

If the above conditions are met, an execution of the **MSR** instruction might not update the destination register correctly. The destination register might contain one of following values after execution:

1. The execution of the **MSR** instruction is ignored. The destination register of the **MSR** instruction holds an old value.
2. The execution of the **MSR** instruction writes an incorrect value to its destination register.

A external debugger and system software are expected to be coordinated to prevent conflict in these registers.

Workaround

No workaround is required for this erratum.

2070947
External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register

Status

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

Description

The core might incorrectly issue a write to External Debug Instruction Transfer Register (EDITR) when an external APB write to another register that is located at offset 0x084 is performed in the Debug state. The following debug components share the offset alias with the EDITR register:

- ETE - TRCVIIECTLR - ViewInst Include/Exclude Control Register
- Reserved locations

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in debug state.
2. The External Debug Status and Control Register (EDSCR) cumulative error flag field is 0b0.
3. Memory access mode is disabled, in example, EDSCR.MA = 0b0.
4. The OS Lock is unlocked.
5. External APB write is performed to a memory mapped register at offset 0x084 other than the EDITR.

Implications

If the above conditions are met, then the core might issue a write to the EDITR and try to execute the instruction pointed to by the ITR. As a result of the execution, the following might happen:

- CPU state and/or memory might get corrupted.
- The CPU might generate an UNDEFINED exception.
- The EDSCR.ITE bit will be set to 0.

Workaround

Before programming any register at this offset when the PE is in Debug state, the debugger should either:

- Set the EDSCR.ERR bit by executing some Undefined instruction (e.g. writing zero to EDITR); or
- Set the OS Lock and then unlock it afterwards.

2089669
OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain

Status

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

Description

OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain. If a Warm Reset occurs, then the value in this register will be lost.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Warm Reset is asserted.

Implications

If the above conditions are met, then the value in OSECCR_EL1/EDECCR will be lost.

Workaround

A debugger should enable a Reset Catch debug event by setting EDECR.RCE to 1. This causes the PE to generate a Reset Catch debug event on a Warm reset, allowing the debugger to reprogram the EDECCR.

2134909
MPAM value associated with instruction fetch might be incorrect

Status

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

Description

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

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs under the following condition:

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

Implications

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

Workaround

There is no workaround.

2137418
A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely

Status

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

Description

Executing an A64 WFI or WFE instruction while in Debug state results in suspension of execution, and execution cannot be resumed by the normal WFI or WFE wake-up events while in Debug state.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The Processing Element (PE) is in Debug state and in AArch64 Execution state.
2. A WFI or WFE instruction is executed from EDITR.

Implications

If the above conditions are met, the PE will suspend execution.

This is not thought to be a serious erratum, because an attempt to execute a WFI or WFE instruction while in Debug state is not expected.

For WFI executed in Debug state, execution can only resume by any of the following:

- A Cold or Warm reset
- A Restart request trigger event from the Cross Trigger Interface (CTI) causing exit from Debug state, followed by a WFI wake-up event

For WFE executed in Debug state, execution can only resume by any of the following:

- A Cold or Warm reset
- A Restart request trigger event from the CTI causing exit from Debug state, followed by a WFE wake-up event
- An external event that sets the Event Register. Examples include executing an SEV instruction on another PE in the system or an event triggered by the Generic Timer.

Workaround

A workaround is unnecessary, because an attempt to execute a WFI or WFE instruction while in Debug state is not expected.

2143135
Some SVE PMU events count incorrectly

Status

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

Description

The following Performance Monitoring Unit (PMU) events do not count correctly:

- 0x8074, SVE_PRED_SPEC, SVE predicated Operations speculatively executed
- 0x8075, SVE_PRED_EMPTY_SPEC, SVE predicated operations with no active predicates, Operations speculatively executed
- 0x8076, SVE_PRED_FULL_SPEC, SVE predicated operations with all active predicates, Operations speculatively executed
- 0x8077, SVE_PRED_PARTIAL_SPEC, SVE predicated operations with partially active predicates, Operations speculatively executed

Configurations Affected

This erratum affects all configurations.

Conditions

One of the PMU event counters is configured to count any of the following events:

- 0x8074
- 0x8075
- 0x8076
- 0x8077

Implications

Load and store operations due to SVE instructions are not counted by any of these events. The counter values for these events will only reflect predicated SVE data processing operations. For example, this means that the ratios of each of the 0x8075-0x8077 event values to the 0x8074 event value will not be as expected because load and store operations are not included. However, the types of predicate used by data processing operations will still be usefully indicated.

Workaround

There is no workaround.

2183513
An SError might not be reported for an atomic store that encounters data poison

Status

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

Description

Under certain conditions, an atomic store that encounters data poison might not report an SError.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An atomic store that is unaligned to its data size but within a 16-byte boundary accesses memory.
2. The atomic store accesses multiple L1 data banks such that not all banks have data poison.

Implications

If the above conditions are met, an SError might not be reported although poisoned data is consumed. Note that the data remains poisoned in the L1 and will be reported on the next access.

Workaround

This erratum has no workaround.

2238112
Reads of DISR_EL1 incorrectly return 0s while in Debug State

Status

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

Description

When the *Processing Element* (PE) is in Debug State, reads of DISR_EL1 from EL1 or EL2 with SCR_EL3.EA=0x1 will incorrectly return 0s.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The PE is executing in Debug State at EL1 or EL2, with SCR_EL3.EA=0x1.
2. The PE executes an MRS to DISR_EL1.

Implications

If the above conditions are met, then the read of DISR_EL1 will incorrectly return 0s.

Workaround

No workaround is expected to be required.

2239141
DRPS instruction is not treated as UNDEFINED at EL0 in Debug state

Status

Fault Type: Programmer Category C.
Fault Status: Present in r0p1, r0p2. Open.

Description

In Debug state, DRPS is not treated as an UNDEFINED instruction.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The *Processing Element* (PE) is in Debug state.
2. PE is executing at EL0.
3. PE executes DRPS instruction.

Implications

If the above conditions are met, then the PE will incorrectly execute DRPS as NOP instead of treating it as an UNDEFINFED instruction.

Workaround

There is no workaround.

2261535
L1 Data poison is not cleared by a store

Status

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

Description

The L1 Data poison is not cleared by a store under certain conditions.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. A Processing Element (PE) executes a store that does not write a full word to a location that has data marked as poison.
2. The PE executes another store that writes to all bytes that contain data poison before the previous store is globally observable.

Implications

If the above conditions are met, then the poison bit in the L1 Data cache does not get cleared.

Workaround

This erratum can be avoided by inserting a DMB before and after a word-aligned store that is intended to clear the poison bit.

2262865
Incorrect sampling of SPE event "Partial predicate" for SVE instruction with no vector operands

Status

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

Description

Under certain circumstances, the SPE event E[17] "Partial predicate" might not be captured as required.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. SPE samples an SVE instruction with no vector operands
2. Governing predicate is "all-false" (empty)

Implications

If the above conditions are met, then the SPE event E[17] "Partial predicate" might not be captured for the given instruction.

Workaround

There is no workaround.

2276445
SPE, PMU event for full/partial/empty/not full predicate might be incorrect for some SVE instructions

Status

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

Description

The SPE, PMU events for full/partial/empty/not full predicate capture the cases where an instruction reads a full, not full, partial, or empty value for governing predicate according to the size of the instruction. Under certain circumstances, the event might be incorrectly captured.

Configurations Affected

This erratum affects all configurations.

Conditions

- PMU is configured to sample events for SVE_PRED_EMPTY_SPEC (0x8075), SVE_PRED_FULL_SPEC (0x8076), SVE_PRED_NOT_FULL_SPEC (0x8079), or SVE_PRED_PARTIAL_SPEC (0x8077).
- One of these SVE conversion instructions is executed: SCVTF, UCVTF, FCTVZU, FCVTZS, FCVT, FCVTX, FCVTXNT, or FCVTNT.
- Governing predicate used by instruction has a different value than All-Active or All-Empty.

or

- SPE samples one of these SVE conversion instructions: SCVTF, UCVTF, FCTVZU, FCVTZS, FCVT, FCVTX, FCVTXNT, or FCVTNT.
- Governing predicate used by instruction has a different value than All-Active or All-Empty.

Implications

If the previous conditions are met, the following events might be incorrectly captured:

- SPE event E[17] "Partial predicate"
- SPE event E[18] "Empty predicate"
- PMU event SVE_PRED_EMPTY_SPEC (0x8075)
- PMU event SVE_PRED_FULL_SPEC (0x8076)
- PMU event SVE_PRED_NOT_FULL_SPEC (0x8079)
- PMU event SVE_PRED_PARTIAL_SPEC (0x8077)

Workaround

This erratum has no workaround.

2278133
PMU L1D_CACHE_REFILL_OUTER is inaccurate

Status

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

Description

The L1D_CACHE_REFILL_OUTER PMU event 0x45 is inaccurate due to ignoring refills generated from a system cache. The L1D_CACHE_REFILL PMU event 0x3 should be the sum of PMU events L1D_CACHE_REFILL_INNER 0x44 and L1D_CACHE_REFILL_OUTER 0x45, however, due to the inaccuracy of L1D_CACHE_REFILL_OUTER 0x45 it is possible that this might not be the case.

Note: L1D_CACHE_REFILL PMU event 0x3 does accurately count all L1D cache refills, including refills from a system cache.

Configurations Affected

This erratum affects all configurations which implement a system cache.

Conditions

This erratum occurs under the following conditions:

1. The L2 inner cache is allocated with data transferred from a system cache.

Implications

When the previous condition is met, the L1D_CACHE_REFILL_OUTER PMU event 0x45 does not increment properly.

Workaround

The correct value of L1D_CACHE_REFILL_OUTER PMU event 0x45 can be calculated by subtracting the value of L1D_CACHE_REFILL_INNER PMU event 0x44 from L1D_CACHE_REFILL PMU event 0x3.

2285233
Lower priority exception might be reported when abort condition is detected at both stages of translation

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions apply:

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

Implications

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

Workaround

This erratum has no workaround.

2307835
ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk

Status

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

Description

When a data double bit error or external abort is encountered during a translation table walk, a synchronous exception is reported with the ISV bit set in the ESR_ELx register.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following condition:

1. A data double bit error or external abort is encountered during a translation table walk, and a synchronous exception is reported.

Implications

If the previous condition is met, the ESR_ELx.ISV bit will be set. The ESR[23:14] bits are set with the correct syndrome for the instruction making the access. That is SAS, SSE, SRT, SF, and AR are all set according to the instruction.

Workaround

This erratum has no workaround.

2391681
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, r1p0, r1p1, r1p2. Open.

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.

2444422
PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly

Status

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

Description

The following Performance Monitoring Unit (PMU) events do not count correctly:

- 0x3D, STALL_SLOT_BACKEND, no operation sent for execution on a slot due to the backend
- 0x3E, STALL_SLOT_FRONTEND, no operation sent for execution on a slot due to the frontend

Configurations Affected

This erratum affects all configurations.

Conditions

One of the PMU event counters is configured to count any of the following events:

- 0x3D, STALL_SLOT_BACKEND
- 0x3E, STALL_SLOT_FRONTEND

Implications

When operations are stalled in the processing element's dispatch pipeline slot, some of those slot stalls are counted as frontend stalls when they should have been counted as backend stalls, rendering PMU events 0x3D (STALL_SLOT_BACKEND) and 0x3E (STALL_SLOT_FRONTEND) inaccurate. The PMU event 0x3F (STALL_SLOT) does still accurately reflect its intended count of "No operation sent for execution on a slot".

Workaround

This erratum has no workaround.

2647275
Incorrect read value for Performance Monitors Control Register

Status

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

Description

The Performance Monitors Control Register (PMCR_EL0) and the External Performance Monitor Control Register (PMCR) might return an incorrect read value for the X field.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Software writes a nonzero value to the PMCR_EL0.X, or debugger writes a nonzero value to the PMCR.X
2. Software reads the PMCR_EL0 register, or debugger reads the PMCR register

Implications

The PMCR_EL1.X or PMCR.X field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

Workaround

This erratum has no workaround.

2707725
Incorrect read value for Performance Monitors Configuration Register EX field

Status

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

Description

The Performance Monitors Configuration Register (PMCFGR) might return an incorrect read value for the EX field.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the software reads the PMCFGR register.

Implications

The PMCFGR.EX field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

Workaround

This erratum has no workaround.

2755353
Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

Status

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

Description

Under certain conditions the SAMPLE_POP PMU event 0x4000 might continue to count after SPE profiling has been disabled.

Configurations Affected

This erratum affects all configurations.

Conditions

1. *Statistical Profiling Extension* (SPE) sampling is enabled.
2. *Performance Monitoring Unit* (PMU) event counting is enabled.
3. SPE buffer is disabled, either directly by software, or indirectly via assertion of PMBIRQ, or by entry into Debug state.

Implications

If the previous conditions are met, then the SAMPLE_POP event might reflect an overcounted value. The impact of this erratum is expected to be very minor for actual use cases, as SPE sampling analysis is typically performed independently from PMU event counting.

Workaround

If a workaround is desired, then minimization of potential overcounting of the SAMPLE_POP event can be realized via software disable of any PMU SAMPLE_POP event counters whenever SPE is disabled, and also upon the servicing of a PMBIRQ interrupt. For profiling of EL0 workloads, software can further reduce exposure to overcounting by configuring the counter to not count at Exception levels of EL1 or higher.

2795125
DGH instruction doesn't execute correctly

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. A DGH instruction is executed.

Implications

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

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

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

Workaround

No workaround is expected to be necessary.

2798806
Incorrect decoding of SVE version of PRF* scalar plus scalar instructions

Status

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

Description

Scalar plus Scalar forms of the *Scalable Vector Extension* (SVE) PRF may not prefetch from the correct address. The address should be Xn + Xm << scalar, but is instead calculated as Xn. This affects the following instructions:

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. Any of the above instructions are executed without trapping when Xm != 0x0

Implications

All affected instructions are software prefetches which do not affect architectural state in any way (including suppression of any translation faults). Thus this erratum will not affect the functional operation of the CPU. Since these instructions are likely to be used in contexts where Xn is fixed and Xm is incrementing, it is unlikely that the erroneous prefetches would result in undesired cache pollution or reduction in memory bandwidth because the instructions will simply continuously prefetch the same address.

Workaround

No workaround is expected to be necessary, but if one is specifically needed, the programmer can use an ADD, and then one of the immediate forms of SVE PRF, which are unaffected. These instructions are:

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

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

2816902
PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM

Status

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

Description

Under certain conditions, the *Processing Element* (PE) might fail to report multiple uncorrectable *Error Correction Code* (ECC) errors that occur in the L1 data cache tag RAM.

Configurations affected

This erratum affects all configurations.

Conditions

1. The PE detects and reports an uncorrectable ECC error in the L1 data cache tag RAM.
2. The PE detects a second uncorrectable ECC error in the L1 data cache tag RAM and an uncorrectable ECC error in the L1 data cache data RAM.

Implications

If the previous conditions are met, then the PE might fail to report the second uncorrectable ECC error in the L1 data cache tag RAM and the address recorded in ERR0ADDR might have an incorrect value. The ECC error occurring in the L1 data cache data RAM is reported correctly.

Workaround

No workaround is necessary. This erratum represents a condition where multiple uncorrectable ECC errors occur in a short period of time. While the PE does not report the errors correctly, ECC still provides a valuable mechanism for error detection and correction.

2910962 L2D_CACHE_WB_CLEAN overcounts

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

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

Implications

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

Workaround

No workaround is required for this erratum.

2985982
SPE latency counters are corrupted under certain conditions

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround.

3605044
Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

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

Implications

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

Workaround

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

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

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

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

Arm Architecture Reference Manual relevant events:

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

Implementation Defined relevant event:

Mnemonic	Number
L1D_TLB_REFILL_RD_PF	0x010E

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

3607341
PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following condition applies:

- MSR PSTATE.{PAN or UAO} executes

Implications

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

Workaround

A workaround is not expected to be required.

3627242
PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum has no workaround.

3633452
EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

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

Implications

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

Workaround

There is no workaround.

3640938
SPE operation type is corrupted under certain conditions

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

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

Implications

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

Workaround

There is no workaround.

3694440
LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

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

Implications

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

Workaround

No workaround is expected to be necessary for this erratum.

3694464
FFR might not capture the lowest faulting memory element

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

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

Implications

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

Workaround

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

3700174
PE might fail to log a RAS error for L2 data RAM ECC errors

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

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

Implications

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

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

Workaround

No workaround is necessary for this erratum.

3705910
PMU events are mis-categorized by not considering the effect of "Taken locally"

Status

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

Description

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

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

Configurations affected

This erratum affects all configurations.

Conditions

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

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

Implications

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

Workaround

There is no workaround.

Proprietary notice

(PRE-1121-V1.0)

Product and document information

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

Product status

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

Product completeness status

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

Product revision status

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

rx

Identifies the major revision of the product.

py

Identifies the minor revision or modification status of the product.