Arm Neoverse N1 (MP050)

Software Developer Errata Notice

Date of issue: October 10, 2024

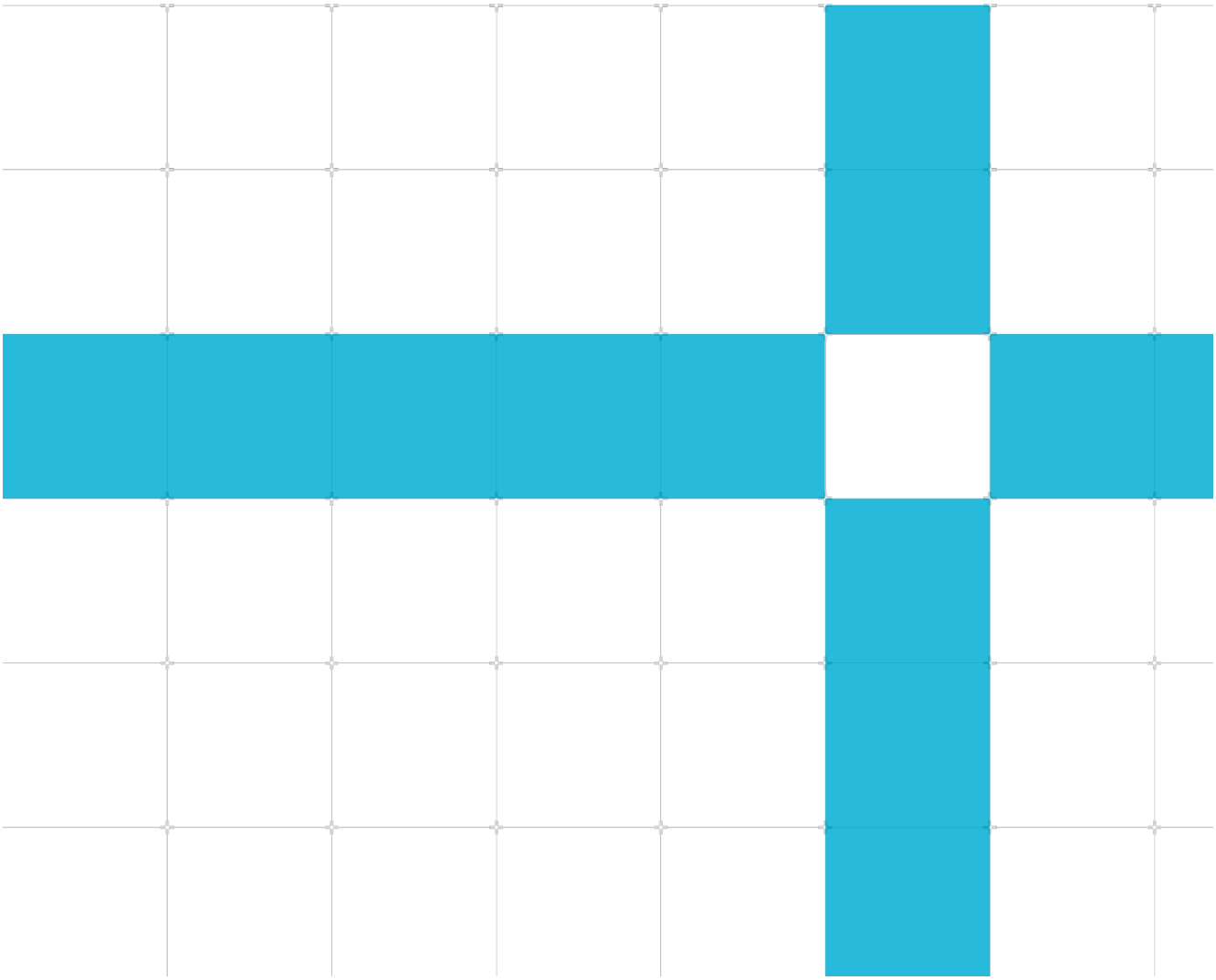
Non-Confidential

Document version: 33.0

Copyright © 2024 Arm® Limited (or its affiliates). All rights reserved.

Document ID: SDEN-885747

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



This document is Non-Confidential.

Copyright © 2024 Arm® Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted Arm's Proprietary notice found at the end of this document.

This document (SDEN_885747_33.0_en) was issued on October 10, 2024.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-885747**

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 N1 (MP050), 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

 Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

 Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

 Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

r1p0 implementation fixes

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

REVIDR_EL1[0]	1043202 AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check

Note that there is no change to the MIDR_EL1 which remains at r1p0 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

r2p0 implementation fixes

Note the following errata might be fixed in some implementations of r2p0. 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]	1220197 Streaming store under specific conditions might cause deadlock or data corruption

Note that there is no change to the MIDR_EL1 which remains at r2p0 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

r3p1 implementation fixes

Note the following errata might be fixed in some implementations of r3p1. 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[3]	1349291 Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError
REVIDR_EL1[4]	1354823 SnpOnceFwd might return incorrect data
REVIDR_EL1[5]	1356341 L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events
REVIDR_EL1[6]	1415323 Ordering violation might occur when a load encounters a L1 tag RAM single bit ECC error in the presence of a snoop request targeting the same line

Note that there is no change to the MIDR_EL1 which remains at r3p1 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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 10, 2024: Changes in document version v33.0

ID	Status	Area	Category	Summary
3696297	New	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock
3605051	New	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed
3607350	New	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative
3633468	New	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception
3700183	New	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors
3705920	New	Programmer	Category C	PMU events are mis-categorized by not considering the effect of "Taken locally"

May 09, 2024: Changes in document version v32.0

ID	Status	Area	Category	Summary
3324349	New	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing

September 28, 2023: Changes in document version v31.0

ID	Status	Area	Category	Summary
3023823	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
1286807	Updated	Programmer	Category B (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation
2486423	New	Programmer	Category C	L1D_TLB access related PMU event increments more than once per memory access
2910961	New	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts

April 26, 2023: Changes in document version v30.0

ID	Status	Area	Category	Summary
1257314	Updated	Programmer	Category B	Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption or deadlock

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

January 31, 2023: Changes in document version v29.0

ID	Status	Area	Category	Summary
1262888	Updated	Programmer	Category B	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation
2729172	New	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP
2816904	New	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM

October 14, 2022: Changes in document version v28.0

ID	Status	Area	Category	Summary
1262888	Updated	Programmer	Category B	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation
2743102	New	Programmer	Category B	The core might deadlock during powerdown sequence

February 08, 2022: Changes in document version v27.0

ID	Status	Area	Category	Summary
1458230	Updated	Programmer	Category B	Software Step might prevent interrupt recognition
2356586	New	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
2227007	Updated	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate
2307838	New	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk
2391683	New	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR

August 16, 2021: Changes in document version v26.0

ID	Status	Area	Category	Summary
1850713	Updated	Programmer	Category B	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR
2110726	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
2141647	New	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely
2153915	New	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes
2227007	New	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate
2238117	New	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State
2239143	New	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state
2263697	New	Programmer	Category C	L1 Data poison is not cleared by a store

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

March 02, 2021: Changes in document version v25.0

ID	Status	Area	Category	Summary
2019409	New	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom
2052428	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

November 06, 2020: Changes in document version v24.0

ID	Status	Area	Category	Summary
1978083	New	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock
2001418	New	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL
2001723	New	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled

September 29, 2020: Changes in document version v23.0

ID	Status	Area	Category	Summary
1923202	New	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset
1946160	New	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics
1930283	New	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

July 31, 2020: Changes in document version v22.0

ID	Status	Area	Category	Summary
1165522	Updated	Programmer	Category B	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation
1800710	Updated	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB
1850713	New	Programmer	Category B	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR
1868343	New	Programmer	Category B	The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary
1857203	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
1857622	New	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock
1874565	New	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect
1880110	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events
1899209	New	Programmer	Category C	Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO
1899433	New	Programmer	Category C	PFG duplicate reported faults through a Warm reset
1912195	New	Programmer	Category C	SPE events for "Other" operation type records might be captured incorrectly
1913776	New	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors

May 22, 2020: Changes in document version v21.0

ID	Status	Area	Category	Summary
1791580	New	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures
1800710	New	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB
1779123	New	Programmer	Category C	External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior
1788066	New	Programmer	Category C	Possible loss of CTI event
1788068	New	Programmer	Category C	Loss of CTI events during warm reset
1814889	New	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

February 14, 2020: Changes in document version v20.0

ID	Status	Area	Category	Summary
1688568	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
1688567	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
1655746	New	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption
1694299	New	Programmer	Category C	Instruction sampling bias exists in SPE implementation

December 20, 2019: Changes in document version v19.0

ID	Status	Area	Category	Summary
1627784	New	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect
1662732	New	Programmer	Category C	Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior
1697035	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

September 27, 2019: Changes in document version v18.0

ID	Status	Area	Category	Summary
1542419	Updated	Programmer	Category B (rare)	The core might fetch a stale instruction from memory which violates the ordering of instruction fetches

September 24, 2019: Changes in document version v17.0

ID	Status	Area	Category	Summary
1533195	New	Programmer	Category B	Accessing a memory location using mismatched shareability attributes might cause loss of coherency
1542419	New	Programmer	Category B (rare)	The core might fetch a stale instruction from memory which violates the ordering of instruction fetches
1514034	New	Programmer	Category C	Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE
1523502	New	Programmer	Category C	CPUECTLR_EL1 controls for the MMU have no affect

June 28, 2019: Changes in document version v16.0

ID	Status	Area	Category	Summary
1467587	New	Programmer	Category B	HCR_EL2.TOCU incorrectly applies during EL0 execution when HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1
1346756	Updated	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
1487185	New	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled
1490853	New	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

April 30, 2019: Changes in document version v15.0

ID	Status	Area	Category	Summary
1354823	Updated	Programmer	Category B	SnpOnceFwd might return incorrect data
1458230	New	Programmer	Category B	Software Step might prevent interrupt recognition
1418040	Updated	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
1349291	Updated	Programmer	Category C	Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError
1356341	Updated	Programmer	Category C	L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events
1395332	Updated	Programmer	Category C	Read from PMCCNTR in AArch32 might return corrupted data
1406411	Updated	Programmer	Category C	MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit
1408724	Updated	Programmer	Category C	Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address
1415323	Updated	Programmer	Category C	Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line
1430754	New	Programmer	Category C	Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences

March 26, 2019: Changes in document version v14.0

ID	Status	Area	Category	Summary
1418040	New	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
1406411	New	Programmer	Category C	MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit
1408724	New	Programmer	Category C	Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address
1415323	New	Programmer	Category C	Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

March 08, 2019: Changes in document version v13.0

ID	Status	Area	Category	Summary
1354823	New	Programmer	Category B	SnpOnceFwd might return incorrect data
1346756	New	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
1349291	New	Programmer	Category C	Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError
1356341	New	Programmer	Category C	L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events
1395332	New	Programmer	Category C	Read from PMCCNTR in AArch32 might return corrupted data

November 21, 2018: Changes in document version v12.0

ID	Status	Area	Category	Summary
1315703	New	Programmer	Category A (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-write ordering violation
1257314	Updated	Programmer	Category B	Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption or deadlock
1262606	Updated	Programmer	Category B	Concurrent instruction TLB miss and mispredicted branch instruction located at the end of 32MB region might fetch wrong instruction stream
1262888	Updated	Programmer	Category B	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation
1275112	Updated	Programmer	Category B	A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock
1286807	New	Programmer	Category B (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation
1227053	Updated	Programmer	Category C	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering
1244984	Updated	Programmer	Category C	Illegal return event might corrupt PSTATE.UA0
1256788	Updated	Programmer	Category C	Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error
1264383	Updated	Programmer	Category C	Write-Back load after two Device-nG* stores to the same physical address might get invalid data

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

October 04, 2018: Changes in document version v11.0

ID	Status	Area	Category	Summary
1257314	New	Programmer	Category B	Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption or deadlock
1262606	New	Programmer	Category B	Concurrent instruction TLB miss and mispredicted branch instruction located at the end of 32MB region might fetch wrong instruction stream
1262888	New	Programmer	Category B	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation
1275112	New	Programmer	Category B	A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock
1227053	New	Programmer	Category C	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering
1244984	New	Programmer	Category C	Illegal return event might corrupt PSTATE.UA0
1256788	New	Programmer	Category C	Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error
1264383	New	Programmer	Category C	Write-Back load after two Device-nG* stores to the same physical address might get invalid data

September 07, 2018: Changes in document version v10.0

ID	Status	Area	Category	Summary
1165522	Updated	Programmer	Category B	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation
1188873	Updated	Programmer	Category B	MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result
1220197	New	Programmer	Category B	Streaming store under specific conditions might cause deadlock or data corruption

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

August 01, 2018: Changes in document version v9.0

ID	Status	Area	Category	Summary
1130799	New	Programmer	Category B	TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data
1165347	New	Programmer	Category B	Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock
1165522	New	Programmer	Category B	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation
1188873	New	Programmer	Category B	MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result
1207823	New	Programmer	Category B	The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence
1096402	Updated	Programmer	Category C	Exception packet for return stack match might return incorrect [E1:E0] field
1109624	Updated	Programmer	Category C	Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock
1119735	Updated	Programmer	Category C	16-bit T32 instruction close to breakpoint location might cause early breakpoint exception
1126105	Updated	Programmer	Category C	Read from L1 instruction cache data array using RAMINDEX operation might return data from the wrong location
1144394	Updated	Programmer	Category C	Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error
1192279	New	Programmer	Category C	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level
1194748	Updated	Programmer	Category C	The ERXADDR_EL1 register might report an incorrect physical address for an L1 data tag RAM single-bit correctable ECC error
1194749	Updated	Programmer	Category C	ERR0MISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array
1214504	New	Programmer	Category C	Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry
1227629	New	Programmer	Category C	ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

June 22, 2018: Changes in document version v8.0

ID	Status	Area	Category	Summary
1096402	New	Programmer	Category C	Exception packet for return stack match might return incorrect [E1:E0] field
1109624	New	Programmer	Category C	Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock
1119735	New	Programmer	Category C	16-bit T32 instruction close to breakpoint location might cause early breakpoint exception
1126105	New	Programmer	Category C	Read from L1 instruction cache data array using RAMINDEX operation might return data from the wrong location
1144394	New	Programmer	Category C	Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error
1194748	New	Programmer	Category C	The ERXADDR_EL1 register might report an incorrect physical address for an L1 data tag RAM single-bit correctable ECC error
1194749	New	Programmer	Category C	ERR0MISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array

April 13, 2018: Changes in document version v7.0

ID	Status	Area	Category	Summary
1039219	Updated	Programmer	Category B	When using SPE, sampling certain system register instructions might cause deadlock
1043202	Updated	Programmer	Category B	AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check
1073348	New	Programmer	Category B	Concurrent instruction TLB miss and mispredicted return instruction might fetch wrong instruction stream
1051464	Updated	Programmer	Category C	CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed
1057923	New	Programmer	Category C	Extra instruction might be executed during Halting Step when stepping WFI, WFE and some self-synchronizing system register writes
1069401	New	Programmer	Category C	Debug APB accesses to the ELA RAM might return incorrect data

April 03, 2018: Changes in document version v6.0

No new or updated errata in this document version.

February 09, 2018: Changes in document version v5.0

ID	Status	Area	Category	Summary
1039219	New	Programmer	Category B	When using SPE, sampling certain system register instructions might cause deadlock
1043202	New	Programmer	Category B	AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check
1051464	New	Programmer	Category C	CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

October 23, 2017: Changes in document version v4.0

ID	Status	Area	Category	Summary
925373	New	Programmer	Category B	Executing a WFx instruction while SPE is enabled might cause deadlock
977072	New	Programmer	Category B	Accessing certain Debug or Generic Timer system registers in AArch32 might cause incorrect system register values
981980	New	Programmer	Category B	Interrupt is taken immediately after MSR DAIF instruction masks the interrupt
930017	New	Programmer	Category C	Failure to sign-extend instruction virtual address when using the SPE
937437	New	Programmer	Category C	An SPE buffer full event might clear PMBSR_EL1.DL and PMBSR_EL1.EA
941868	New	Programmer	Category C	Deferred errors might cause silent data corruption following a hardware update of Access and Dirty bits in a translation table entry
944783	New	Programmer	Category C	Address breakpoint might cause a deadlock with certain AArch32 T32 code sequences
961111	New	Programmer	Category C	L2 might report multiple RAS errors for the same prefetch request
964384	New	Programmer	Category C	Stuck-at-fault in L1 instruction cache data array might cause deadlock with certain AArch32 T32 code sequences
978245	New	Programmer	Category C	Executing unallocated encoding in conversion between floating-point and integer instruction class does not generate Undefined Instruction exception
986709	New	Programmer	Category C	MRS to DBGDTR_EL0 might cause EDSCR.RXfull bit to clear incorrectly
988575	New	Programmer	Category C	Unaligned cache line split load to NC or Device memory, tagged with poison or external error on its first half, might cause data corruption

September 25, 2017: Changes in document version v3.0

ID	Status	Area	Category	Summary
931711	New	Programmer	Category B	Reads from DSU CLUSTER* or ERX* system registers might return corrupted data
933092	New	Programmer	Category C	Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort
933779	New	Programmer	Category C	DBGDTRTX register fails to hold value through Warm reset
934968	New	Programmer	Category C	DCPSx instruction with SCTLR_EL1.IESB = 1 while in debug state might not execute correctly

August 18, 2017: Changes in document version v2.0

ID	Status	Area	Category	Summary
905797	New	Programmer	Category B	Failure to enforce read-after-read ordering rules
901361	New	Programmer	Category C	Failure to report or incorrect reporting of L2 data RAM ECC errors
901865	New	Programmer	Category C	Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock
902290	New	Programmer	Category C	Persistent error response to transactions issued on behalf of Page descriptor Access bit and Dirty bit updates might livelock
909055	New	Programmer	Category C	Failure to record L1 data cache access event when using the SPE

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

May 29, 2017: Changes in document version v1.0

No errata in this document version.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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
1315703	Programmer	Category A (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-write ordering violation	r0p0, r1p0, r2p0, r3p0	r3p1
905797	Programmer	Category B	Failure to enforce read-after-read ordering rules	r0p0	r1p0
925373	Programmer	Category B	Executing a WFx instruction while SPE is enabled might cause deadlock	r0p0	r1p0
931711	Programmer	Category B	Reads from DSU CLUSTER* or ERX* system registers might return corrupted data	r0p0	r1p0
977072	Programmer	Category B	Accessing certain Debug or Generic Timer system registers in AArch32 might cause incorrect system register values	r0p0	r1p0
981980	Programmer	Category B	Interrupt is taken immediately after MSR DAIF instruction masks the interrupt	r0p0	r1p0
1039219	Programmer	Category B	When using SPE, sampling certain system register instructions might cause deadlock	r0p0, r1p0	r2p0
1043202	Programmer	Category B	AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check	r0p0, r1p0	r2p0
1073348	Programmer	Category B	Concurrent instruction TLB miss and mispredicted return instruction might fetch wrong instruction stream	r0p0, r1p0	r2p0
1130799	Programmer	Category B	TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data	r0p0, r1p0, r2p0	r3p0
1165347	Programmer	Category B	Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock	r0p0, r1p0, r2p0	r3p0

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
1165522	Programmer	Category B	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation	r0p0, r1p0, r2p0	r3p0
1188873	Programmer	Category B	MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result	r0p0, r1p0, r2p0	r3p0
1207823	Programmer	Category B	The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence	r0p0, r1p0, r2p0	r3p0
1220197	Programmer	Category B	Streaming store under specific conditions might cause deadlock or data corruption	r0p0, r1p0, r2p0	r3p0
1257314	Programmer	Category B	Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption or deadlock	r0p0, r1p0, r2p0, r3p0	r3p1
1262606	Programmer	Category B	Concurrent instruction TLB miss and mispredicted branch instruction located at the end of 32MB region might fetch wrong instruction stream	r0p0, r1p0, r2p0, r3p0	r3p1
1262888	Programmer	Category B	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation	r0p0, r1p0, r2p0, r3p0	r3p1
1275112	Programmer	Category B	A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock	r0p0, r1p0, r2p0, r3p0	r3p1
1354823	Programmer	Category B	SnpOnceFwd might return incorrect data	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1458230	Programmer	Category B	Software Step might prevent interrupt recognition	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1467587	Programmer	Category B	HCR_EL2.TOCU incorrectly applies during EL0 execution when HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1533195	Programmer	Category B	Accessing a memory location using mismatched shareability attributes might cause loss of coherency	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
1688567	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, r2p0, r3p0, r3p1, r4p0	r4p1
1688568	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, r2p0, r3p0, r3p1, r4p0	r4p1
1791580	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1800710	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, r2p0, r3p0, r3p1, r4p0	r4p1
1850713	Programmer	Category B	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1868343	Programmer	Category B	The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1923202	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
1946160	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
1978083	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2356586	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2743102	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3023823	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
3324349	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3696297	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	Open	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1
1286807	Programmer	Category B (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation	r0p0, r1p0, r2p0, r3p0	r3p1
1418040	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1542419	Programmer	Category B (rare)	The core might fetch a stale instruction from memory which violates the ordering of instruction fetches	r3p0, r3p1, r4p0	r4p1
901361	Programmer	Category C	Failure to report or incorrect reporting of L2 data RAM ECC errors	r0p0	r1p0
901865	Programmer	Category C	Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock	r0p0	r1p0
902290	Programmer	Category C	Persistent error response to transactions issued on behalf of Page descriptor Access bit and Dirty bit updates might livelock	r0p0	r1p0
909055	Programmer	Category C	Failure to record L1 data cache access event when using the SPE	r0p0	r1p0
930017	Programmer	Category C	Failure to sign-extend instruction virtual address when using the SPE	r0p0	r1p0
933092	Programmer	Category C	Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort	r0p0	r1p0
933779	Programmer	Category C	DBGDTRTX register fails to hold value through Warm reset	r0p0	r1p0
934968	Programmer	Category C	DCPSx instruction with SCTLR_EL1.IESB = 1 while in debug state might not execute correctly	r0p0	r1p0
937437	Programmer	Category C	An SPE buffer full event might clear PMBSR_EL1.DL and PMBSR_EL1.EA	r0p0	r1p0

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
941868	Programmer	Category C	Deferred errors might cause silent data corruption following a hardware update of Access and Dirty bits in a translation table entry	r0p0	r1p0
944783	Programmer	Category C	Address breakpoint might cause a deadlock with certain AArch32 T32 code sequences	r0p0	r1p0
961111	Programmer	Category C	L2 might report multiple RAS errors for the same prefetch request	r0p0	r1p0
964384	Programmer	Category C	Stuck-at-fault in L1 instruction cache data array might cause deadlock with certain AArch32 T32 code sequences	r0p0	r1p0
978245	Programmer	Category C	Executing unallocated encoding in conversion between floating-point and integer instruction class does not generate Undefined Instruction exception	r0p0	r1p0
986709	Programmer	Category C	MRS to DBGDTR_EL0 might cause EDSCR.RXfull bit to clear incorrectly	r0p0	r1p0
988575	Programmer	Category C	Unaligned cache line split load to NC or Device memory, tagged with poison or external error on its first half, might cause data corruption	r0p0	r1p0
1051464	Programmer	Category C	CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed	r0p0, r1p0	r2p0
1057923	Programmer	Category C	Extra instruction might be executed during Halting Step when stepping WFI, WFE and some self-synchronizing system register writes	r0p0, r1p0	r2p0
1069401	Programmer	Category C	Debug APB accesses to the ELA RAM might return incorrect data	r0p0, r1p0	r2p0
1096402	Programmer	Category C	Exception packet for return stack match might return incorrect [E1:E0] field	r0p0, r1p0, r2p0	r3p0
1109624	Programmer	Category C	Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock	r0p0, r1p0, r2p0	r3p0
1119735	Programmer	Category C	16-bit T32 instruction close to breakpoint location might cause early breakpoint exception	r0p0, r1p0, r2p0	r3p0
1126105	Programmer	Category C	Read from L1 instruction cache data array using RAMINDEX operation might return data from the wrong location	r0p0, r1p0, r2p0	r3p0

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
1144394	Programmer	Category C	Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error	r0p0, r1p0, r2p0	r3p0
1192279	Programmer	Category C	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level	r0p0, r1p0, r2p0	r3p0
1194748	Programmer	Category C	The ERXADDR_EL1 register might report an incorrect physical address for an L1 data tag RAM single-bit correctable ECC error	r0p0, r1p0, r2p0	r3p0
1194749	Programmer	Category C	ERR0MISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array	r0p0, r1p0, r2p0	r3p0
1214504	Programmer	Category C	Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry	r0p0, r1p0, r2p0	r3p0
1227053	Programmer	Category C	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering	r0p0, r1p0, r2p0, r3p0	r3p1
1227629	Programmer	Category C	ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported	r0p0, r1p0, r2p0	r3p0
1244984	Programmer	Category C	Illegal return event might corrupt PSTATE.UA0	r0p0, r1p0, r2p0, r3p0	r3p1
1256788	Programmer	Category C	Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error	r0p0, r1p0, r2p0, r3p0	r3p1
1264383	Programmer	Category C	Write-Back load after two Device-nG* stores to the same physical address might get invalid data	r0p0, r1p0, r2p0, r3p0	r3p1
1346756	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
1349291	Programmer	Category C	Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1356341	Programmer	Category C	L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
1395332	Programmer	Category C	Read from PMCCNTR in AArch32 might return corrupted data	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1406411	Programmer	Category C	MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1408724	Programmer	Category C	Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1415323	Programmer	Category C	Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1430754	Programmer	Category C	Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
1487185	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1490853	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1514034	Programmer	Category C	Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1523502	Programmer	Category C	CPUECTLR_EL1 controls for the MMU have no affect	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1627784	Programmer	Category C	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1655746	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1662732	Programmer	Category C	Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior	r3p0, r3p1, r4p0	r4p1
1694299	Programmer	Category C	Instruction sampling bias exists in SPE implementation	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1697035	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, r2p0, r3p0, r3p1, r4p0	r4p1

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
1779123	Programmer	Category C	External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1788066	Programmer	Category C	Possible loss of CTI event	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1788068	Programmer	Category C	Loss of CTI events during warm reset	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1814889	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1857203	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, r2p0, r3p0, r3p1, r4p0	r4p1
1857622	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1874565	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, r2p0, r3p0, r3p1, r4p0	r4p1
1880110	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
1899209	Programmer	Category C	Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
1899433	Programmer	Category C	PFG duplicate reported faults through a Warm reset	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
1912195	Programmer	Category C	SPE events for "Other" operation type records might be captured incorrectly	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1913776	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0	r4p1
1930283	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2001418	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
2001723	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2019409	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2052428	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, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2110726	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, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2141647	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2153915	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2227007	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2238117	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2239143	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2263697	Programmer	Category C	L1 Data poison is not cleared by a store	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2307838	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2391683	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2486423	Programmer	Category C	L1D_TLB access related PMU event increments more than once per memory access	r0p0, r1p0, r2p0, r3p0, r3p1	r4p0
2729172	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

ID	Area	Category	Summary	Found in versions	Fixed in version
2816904	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
2910961	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3605051	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3607350	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3633468	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3700183	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open
3705920	Programmer	Category C	PMU events are mis-categorized by not considering the effect of "Taken locally"	r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, r4p1	Open

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Errata descriptions

Category A

There are no errata in this category.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Category A (rare)

1315703
Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-write ordering violation

Status

Fault Type: Programmer Category A Rare
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

If a virtual address for a cacheable mapping of a location is being accessed by a core while another core is remapping the virtual address to a new physical page using the recommended break-before-make sequence, then under rare circumstances TLBI+DSB completes before a write using the translation being invalidated has been observed by other observers.

Configurations Affected

The erratum affects all multi-core configurations.

Conditions.

1. Core A has in program order a store (ST1) and a younger load (LD1) to the same cacheable virtual address.
2. Core B marks the associated translation table entry invalid, followed by a DSB; TLBI; DSB sequence which generates a sync request to Core A.
3. LD1 executes speculatively past ST1 and returns its result using the original physical address (PA1) under specific rare conditions before Core A has responded to the sync request.
4. At the time of receiving the sync request, on Core A:
    a. No load younger than ST1 has executed out-of-order for any of the following instructions:
        i. Load.
        ii. DMB.
        iii. DSB.
        iv. Atomic instruction which updates a register and has acquire semantics.
    b. No store younger than ST1 has already computed its physical address (PA).
5. Any memory request from core A which was initiated prior to the sync request completes.
6. ST1 is not able to compute its PA before Core A responds to the sync request.
7. Core B receives the sync response and updates the translation table entry to map a new PA (PA2), which has write permissions and differs on bits [23:12] from PA1, followed by a DSB.
8. ST1 performs memory write using PA2 on Core A and commits the result from LD1 using PA1 because the read-after-write ordering violation between ST1 and LD1 is not detected.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Implications

If the above conditions are met under certain rare conditions, then this erratum might result in a read-after-write ordering violation.

Workaround

This erratum can be avoided by setting PSTATE.SSBS to 0 or CPUACTLR2_EL1[16] to 1, hence preventing LD1 from speculating past ST1. This will have a performance impact on general workloads.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Category B

905797
Failure to enforce read-after-read ordering rules

Status

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

Description

Under a combination of unusual conditions, it is possible for a younger load to bypass an older load to the same address, and for the two loads to observe updates to that address in the wrong order. In other words, the younger load might observe data which is globally ordered before the data that is observed by the older load.

Configurations Affected

This erratum affects all configurations.

Conditions

1. PE 1 is executing a program which contains a load atomic to physical address A, followed by two cacheable loads to physical address B. The load atomic does not have acquire semantics.
2. PE 2 is executing a program which modifies the value stored at physical address B.

If the above conditions are met under certain unusual timing conditions, then it is possible for the older of the two loads on PE 1 to observe the newly modified value from PE 2 while the younger load on PE 1 observes the value before the modification by PE 2.

Implications

If this erratum occurs, then multi-threaded software which relies on the read ordering rules might get an incorrect result.

Workaround

A workaround is not expected to be necessary in most cases. This is because the conditions require a combination of unusual timing conditions and load atomic instructions, which are not yet widely in use.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

However, for systems using an ACE interconnect or a CHI interconnect that does not support far atomic operations (thus input **BROADCASTATOMIC** pin is tied to 0), setting CPUACTRL2_EL1[2] to 1 will prevent the conditions necessary to hit this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

925373
Executing a WFx instruction while SPE is enabled might cause deadlock

Status

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

Description

When Statistical Profiling Extension (SPE) profiling is enabled and physical timestamps are being collected, executing a WFI or WFE instruction might cause the core to deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. SPE profiling is enabled.
2. Physical timestamps are being collected.
3. The core executes a WFI or WFE instruction.

Implications

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

Workaround

This erratum can be avoided by using either of the following workarounds:

- Disabling SPE profiling.
- Setting CPUACTLR3_EL1[33] and CPUACTLR3_EL1[34] to 1. Setting CPUACTLR3_EL1[33] to 1 results in WFE instruction to be executed as a NOP instruction. Setting CPUACTLR3_EL1[34] to 1 results in WFI instruction to be executed as a NOP instruction.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

931711
Reads from DSU CLUSTER* or ERX* system registers might return corrupted data

Status

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

Description

When a system register read from a particular set of system registers is executed speculatively, a subsequent read from the same set of registers might return corrupted data. The registers affected are the DynamIQ Shared Unit (DSU) CLUSTER* control system registers, and the ERX* error system registers when ERRSELR_EL1.SEL=1.

Configurations Affected

This erratum affects configurations in which the DSU is implemented with the Snoop Control Unit (SCU) present.

Conditions

1. For ERX* registers, ERRSELR_EL1.SEL is set to 1.
2. MRS Instruction A targeting a CLUSTER* or ERX* register is speculatively executed.
3. MRS Instruction B targeting a CLUSTER* or ERX* register is executed before the core receives the read response from Instruction A.

Note that the registers in the first MRS read and second MRS read do not have to be the same for this erratum to occur.

Implications

If the above conditions are met, data returned from the MRS instruction B targeting a CLUSTER* or ERX* register might be corrupted.

Workaround

In most cases, this erratum can be avoided by inserting an ISB instruction before the MRS to the CLUSTER* or ERX* system register, as the ISB can prevent the MRS from being speculatively executed.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

977072
Accessing certain Debug or Generic Timer system registers in AArch32 might cause incorrect system register values

Status

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

Description

Conditional MRC/MCR/MRRC/MCRR accesses, or speculative unconditional MRC/MRRC reads, to certain Debug or Generic Timer system registers in AArch32 state can result in incorrect values for these system registers.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing in AArch32 state.
2. A non-exceptional MRC/MCR/MRRC/MCRR access is made to one of the following registers: DBGDTRTXint, DBGDTRRXint, CNTP_TVAL, CNTP_CTL, CNTV_TVAL, CNTV_CTL, CNTPCT, CNTVCT, CNTP_CVAL, or CNTV_CVAL.

Implications

If the above conditions are met, then a read of an affected register might return an incorrect result, and a write of an affected register might occur unconditionally.

Workaround

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. To trap the DBG* accesses, set MDSCR_EL1.TDCC to 1.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

981980
Interrupt is taken immediately after MSR DAIF instruction masks the interrupt

Status

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

Description

When an interrupt arrives during the execution of an instruction that causes the interrupt to be masked, either MSR DAIFSet (immediate) or MSR DAIF (register), in some circumstances the interrupt will be erroneously taken on the instruction immediately following the MSR. Under the simple sequential execution model, that interrupt should not be taken because it has just been masked. In the interrupt handler, SPSR_ELx and ELR_ELx will reflect the fact that the relevant PSTATE.{A,I,F} mask bit was set when the interrupt was taken.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An MSR DAIFSet (immediate) or MSR DAIF (register) instruction that changes the relevant PSTATE.{A,I,F} bit from 0 to 1 is executing.
2. An interrupt arrives during the execution of that MSR instruction and the execution state of the machine is such that the decision to take that interrupt depends on the PSTATE.{A,I,F} bit.
3. The interrupt is taken on the next instruction after the MSR instruction although it is newly-masked and should not be taken.

Implications

If the above conditions are met, then the interrupt is incorrectly taken on the instruction following the MSR. The SPSR_ELx.{A,I,F} bits and ELR_ELx will indicate that the masking MSR executed before taking the interrupt.

Workaround

Generally, it is expected that software will be robust against taking an interrupt immediately after masking it in PSTATE. If not, then a workaround is to subtract 4 from the ELR and clear the relevant mask bit in the SPSR when the interrupt vector is entered erroneously.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1039219
When using SPE, sampling certain system register instructions might cause deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0. Fixed in r2p0.

Description

When Statistical Profiling Extension (SPE) profiling is enabled, sampling certain system register instructions might cause the core to deadlock. The set of system registers affected includes:

- DynamIQ Shared Unit (DSU) CLUSTER* system registers.
- ERX* system registers when ERRSELR_EL1.SEL=1.
- Generic Timer CNT* system registers.
- SYSL instruction to the IMPLEMENTATION DEFINED instruction space for encodings where CRn=c15 and Op1=[0,1,2,6].

Configurations Affected

For CLUSTER* and ERX* system registers, this erratum affects configurations in which the DSU is implemented with the Snoop Control Unit (SCU) present. For CNT* system registers and SYSL instruction, this erratum affects all configurations.

Conditions

1. SPE profiling is enabled.
2. A read from or write to the following system registers that have been selected for sampling:
    - CLUSTER* system registers.
    - ERX* system registers with ERRSELR_EL1=1.
    - CNT* system registers.
    - SYSL instruction to the IMPLEMENTATION DEFINED instruction space for encodings where CRn=c15 and Op1=[0,1,2,6].

Implications

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

Workaround

This erratum can be avoided by disabling SPE profiling.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1043202
AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0. Fixed in r2p0.

Description

The AArch32 T32 CLREX instruction in an IT block will always clear the exclusive monitor, even when the CLREX condition code fails.
Note: The CLREX instruction does not have a condition code outside of an IT block.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing in AArch32 state.
2. A T32 CLREX instruction is in an IT block, and the CLREX fails the condition code check while the exclusive monitor is in the exclusive state.

Implications

If the above conditions are met, then a subsequent store-exclusive might unexpectedly fail.

Workaround

This erratum can be avoided by replacing all T32 CLREX instructions with an ISB instruction. This can be done 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,=0xF3BF8F2F
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFFFFFFF
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x800200071
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
ISB
```


Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1073348
Concurrent instruction TLB miss and mispredicted return instruction might fetch wrong instruction stream

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0. Fixed in r2p0.

Description

When branches are speculatively executed, either the prediction is correct and the following speculative instruction stream is architecturally executed, or the branch is mispredicted and the pipeline is flushed. Under certain conditions, if an unconditional indirect branch is speculatively executed while a translation table walk response is almost complete, then the speculative instruction stream might not be flushed if the branch was incorrectly predicted.

Configurations Affected

This erratum affects all configurations.

Conditions

1. One of the following unconditional indirect branches is speculatively executed where the branch target crosses a 32MB boundary:
   - A64: RET.
   - A32: BX lr; POP {...,pc}; LDMIA r13!, {...,pc}; LDR PC, [SP], #offset.
   - T32: BX lr; POP {...,pc}; LDMIA r13!, {...,pc}; LDR PC, [SP], #offset.
2. A translation table walk response arrives around the time the branch is speculated.
3. When the branch resolves, bits[24:0] of the speculated address must match the actual address.

Implications

If the above conditions are met, then the core might execute the wrong instruction stream.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[6] to 1, which disables static prediction.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1130799
TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

TLBI VAAE1 or TLBI VAALE1 targeting a page within aggregated address translation data in the L2 TLB invalidates the page, but might also corrupt the translation for other pages in the group. A subsequent translation miss request to a different page within the aggregated group might result in incorrect translation.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Full hardware page aggregation is enabled by setting CPUECTLR_EL1[48] to 1.
2. Multiple ASIDs have aggregated address translation data in different ways of the L2 TLB for the same VA range.
3. TLBI VAAE1 or TLBI VAALE1 targeting a page within the aggregated address translation data is executed.

Implications

If the above conditions are met, then the MMU might generate incorrect translation.

Workaround

This erratum can be avoided by setting CPUACTLR2_EL1[59] to 1. Setting CPUACTLR2_EL1[59] to 1 might have a small impact on performance.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1165347
Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

1. There is a loop that has a branch that is consistently mispredicted.
2. There is an atomic instruction outside of the loop that has the same physical address as the exclusive monitor address of another core, within a cache line. The atomic instruction makes a unique request, snooping that cache line from other cores, and opening the exclusive monitor.

Implications

If the above conditions are met, the core might livelock.

Workaround

This erratum can be avoided by setting CPUACTLR2_EL1[0] to 1 and CPUACTLR2_EL1[15] to 1.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1165522
Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

A speculative Address Translation (AT) instruction translates using registers associated with an out-of-context translation regime and caches the resulting translation in the L2 TLB. A subsequent translation request generated when the out-of-context translation regime is current uses the previous cached L2 TLB entry producing an incorrect virtual to physical mapping.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A speculative AT instruction performs a table walk translating virtual address to physical address using registers associated with an out-of-context translation regime.
2. Address translation data generated during the walk is cached in the L2 TLB.
3. The out-of-context translation regime becomes current and a subsequent memory access is translated using previously cached address translation data in the L2 TLB, resulting in an incorrect virtual to physical mapping.

Implications

If the above conditions are met, the resulting translation would be incorrect.

Workaround

When context-switching the register state for an out-of-context translation regime, system software at EL2 or above must ensure that all intermediate states during the context-switch would report a level 0 translation fault in response to an AT instruction targeting the out-of-context translation regime. Note that a workaround is only required if the system software contains an AT instruction as part of an executable page at EL2 or above.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1188873
MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

Under certain internal timing conditions, an MRC instruction that closely follows an MRRC instruction might produce incorrect data when the MRRC is a read of specific Generic Timer system registers in AArch32 state.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing at AArch32 EL0.
2. An MRRC instruction which reads either the CNTPCT, CNTVCT, CNTP_CVAL, or CNTV_CVAL register is executed.
3. An MRC instruction is executed.

Implications

If this erratum occurs, then the destination register of the MRC is incorrect.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1207823
The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

Under certain conditions, the exclusive monitor that tracks the Physical Address (PA) for the exclusive-access sequence, might end up tracking the incorrect way the cache line is in the L1 cache. As a result, a subsequent STREX might get a false pass, even though the cache line was written to by another master.

Configurations Affected

This erratum affects all configurations.

Conditions

1. There is a load preceding the LDREX/STREX loop that has the same PA as the exclusive monitor address, within a cache line. However the load has a different VA, specifically a different VA[13:12] for 64KB L1 cache.
2. The LDREX issues ahead of this older load, misses the L1, and makes a request out to the L2 by allocating a request buffer. The L2 responds to the request for the LDREX, the line is allocated into the L1 cache, but the LDREX is prevented from picking up the response.
3. The older load subsequently misses the L1 and makes a request to the L2, using the same request buffer as that was previously used by the LDREX.
4. If the LDREX now replays, such that it coincides with the L2 response for the older load with the same PA, but a different VA, then it can forward from the L2 response for this load and complete. At this point, the exclusive monitor ends up capturing the way that this VA-aliased load is allocated into the L1, but the correct index that corresponds to the LDREX.
5. The exclusive monitor now ends up tracking the incorrect cache line. If the line was snooped out, it would therefore not transition to the open state.

Implications

If the above conditions are met, then the core might allow a subsequent STREX to pass, even though the LDREX/STREX sequence was not atomic.

Workaround

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

This erratum can be avoided by setting CPUACTLR2_EL1[11] to 1.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1220197
Streaming store under specific conditions might cause deadlock or data corruption

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

Under certain rare conditions, a streaming write of at least 64 consecutive bytes might send only 32 bytes of data from the L1 data cache to higher level caches.

Configurations Affected

The erratum affects all configurations.

Conditions

1. A store to address A is dispatched down a speculative path, before the write stream was engaged.
2. The write stream was engaged for a full cache line write.
3. A younger store instruction with address A is dispatched.

Implications

If the above conditions are met under certain timing conditions, then this erratum might result in deadlock or data corruption.

Workaround

This erratum can be avoided by setting CPUECTLR_EL1[25:24] to 0b11, which disables write streaming to the L2. This will have an impact on performance for streaming workloads.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1257314
Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption or deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

Under certain conditions, two floating-point divide or square root instructions completing back-to-back and concurrently getting flushed by back-to-back branch mispredicts might result in data corruption or deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Two or more concurrently executing floating-point divide and/or square root instructions need to complete in back-to-back cycles.
2. A branch mispredict arrives concurrently with the completion of the first divide. This divide will flush.
3. Another branch mispredict arrives concurrently with the completion of the second divide. This divide will flush.
4. No other floating-point/vector instructions are in the scheduler to be issued.
5. Newly dispatched instructions coincidentally pick up a register resource that was freed up by the last flushed divide.
6. The newly dispatched instruction gets issued before its producer is issued.

Implications

If the above conditions are met, then this erratum might result in data corruption or deadlock.

Workaround

This erratum can be avoided by setting CPUACTLR3_EL1[10] to 1, which prevents parallel execution of divide and square root instructions.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1262606
Concurrent instruction TLB miss and mispredicted branch instruction located at the end of 32MB region might fetch wrong instruction stream

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

When branches are speculatively executed, either the prediction is correct and the following speculative instruction stream is architecturally executed, or the branch is mispredicted and the pipeline is flushed. Under certain conditions, if a branch located at the end of a 32MB region is speculatively executed while a translation table walk response is almost complete, then the speculative instruction stream might not be flushed if the branch was incorrectly predicted.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A branch instruction located at the end of a 32MB region (if the branch address is PC=X[63:0], PC[24:6]=all1) is speculatively executed.
2. The target of the speculatively executed branch belongs to neither the current 32MB region (PC[63:25] = X[63:25]+0) nor the next sequential region (PC[63:25] = X[63:25]+1).
3. A translation table walk response for the other 32MB region arrives around the time the branch is speculated and written into the L1 instruction TLB.
4. When the branch resolves, bits[24:0] of the speculated address must match the actual address.

Implications

If the above conditions are met, then the core might execute the wrong instruction stream.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1, which delays instruction fetch after branch misprediction. This workaround will have a small impact on performance.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1262888
Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

Under specific conditions, an incorrect virtual to physical mapping might happen because the L2 TLB has been corrupted.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Since the last TLBI-ALLE1, TLB entries have been created where the stage1 mapping is larger than the stage2 mapping.
2. The CPU issues or receives a by-VA TLB operation for a VMID that is not used by the current translation regime.
3. Micro-architectural conditions occur.

The instructions affected by condition #3 are: TLBI VAAE1, TLBI VAAE1IS, TLBI VAALE1 and TLBI VAALE1IS.

Implications

If the above conditions are met, then the MMU might generate an incorrect translation.

Workaround

The workaround is to ensure the L2 TLB only contains EL1 or EL0 records for the current VMID, and no EL1 or EL0 records when executing at EL2 or higher.

EL2 and EL3 should execute:

- TLBI ALLE1
- DSB SY

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

as the first instructions when taking an exception from EL1 or EL0.

Where EL2 or EL3 use AT instructions against the EL1 or EL0 regime to produce a physical address from a virtual address, this should be followed by the above TLBI sequence.

Because of erratum #1165522 **Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation** there is a small chance that a speculated AT instruction at EL2 or EL3 creates TLB entries that match condition #1. Arm does not believe this is likely to coincide with condition #3.

The previous workaround of issuing TLBI VMALLE1 on exiting a guest VM did not cover cases where stage2 was disabled at EL1 or EL0, or describe how early the instruction must be issued.

The original workaround of setting CPUECTLR_EL1[51], which disables the MMU hardware prefetcher, will not resolve this issue due to a subsequent erratum #1523502.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1275112
A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0. r2p0, and r3p0. Fixed in r3p1.

Description

The core might hang when it executes a T32 instruction inside an IT block.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A T32 instruction is inside an IT block.
2. Subsequent instructions repeatedly create branch misprediction. Branch predictor misprediction occurs either because:
   a. Address translation is disabled.
   b. The second half of the T32 instruction can be decoded as 16-bit instruction updating R15 (PC).
   c. Branch predictor RAMs have soft errors.
3. Another IT block instruction is fetched from the speculative instruction stream (that is corrected by the above branch misprediction) and executed before the first T32 instruction is retired from pipeline.

Implications

If the above conditions are met, the core might deadlock as the instruction in the IT block does not complete.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1, which delays instruction fetch after branch misprediction. This workaround will have a small impact on performance.

The workaround for this erratum is the same as the workaround for erratum 1262606.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1354823
SnpOnceFwd might return incorrect data

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

When the core processes a SnpOnceFwd that collides with a pending copyback operation, the SnpOnceFwd might provide incorrect data to the requester.

Configurations affected

This erratum affects configurations with only a single core and without a DSU L3 cache and snoop filter and using the CMN-600 interconnect. Such systems are defined as direct connect using the following RTL parameter values:

- NUM_BIG_CORES: 1.
- L3_CACHE: FALSE.
- ACE: FALSE.
- PORTER_SAM: TRUE.
- ACP: FALSE.
- PERIPH_PORT: FALSE.
- ASYNC_BRIDGE: TRUE.

Conditions

1. A ReadOnce, ReadOnceMakeInvalid, or ReadOnceCleanInvalid with ExpCompAck deasserted is issued by a coherent device.
2. HN-F generates a SnpOnceFwd targeting an Ares core.
3. SnpOnceFwd collides with a pending copyback operation.
4. SnpOnceFwd might provide completion data from a buffer entry that has been deallocated and possibly reallocated by another transaction.

Implications

If the above conditions are met, then a ReadOnce, ReadOnceMakeInvalid, or ReadOnceCleanInvalid operation might observe incorrect data.

Workaround

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

For systems using the CMN-600 interconnect, set por_hnf_aux_ctl[6]=1 to disable SnpOnceFwd for ReadOnce. HN-F will use SnpShared instead. ReadOnceMakeInvalid and ReadOnceCleanInvalid use SnpUnique and are not affected. The workaround might cause some performance degradation, the extent of which is highly system and workload dependent.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1458230
Software Step might prevent interrupt recognition

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

The Software Stepping of a system call instruction (SVC, HVC, or SMC) can prevent recognition of subsequent interrupts when Software Stepping is disabled in the exception handler of the system call. Additionally, unconventional code involving the Software Stepping of an MSR instruction that clears the MDSCR_EL1.SS bit (disables Software Step while stepping) can prevent recognition of subsequent interrupts.

Configurations Affected

This erratum affects all configurations.

Conditions:

Case A:

1. Software Step is enabled.
2. The system configuration is (MDSCR_EL1.KDE==1) or (MDSCR_EL1.KDE==0 and HCR_EL2.E2H==1 and (HCR_EL2.TGE==1 or MDCR_EL2.TDE==1)).
3. An ERET with SPSR_ELx.SS==1 is executed to cause the Software Step state machine to enter the active-not-pending state.
4. A system call instruction (SVC, HVC, or SMC) is executed and generates its system call exception (that is, it is not trapped).
5. The exception handler of the system call disables Software Step by clearing MDSCR_EL1.SS or by setting SPSR_ELx.D such that, upon return, no Software Step exception is taken. Alternatively, under a denial of service attack scenario, the exception handler of the system call continually executes code with no return and thus prevents the taking of a Software Step exception.

Case B:

1. Software Step is enabled.
2. An ERET with SPSR_ELx.SS==1 is executed to cause the Software Step state machine to enter the active-not-pending state.
3. An MSR MDSCR_EL1 instruction that clears the MDSCR_EL1.SS bit is executed (disables Software Step).

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Implications

If either set of the above conditions are met, then interrupts might not be recognized, leading to a denial of service until Software Step is re-enabled and some instruction is stepped that results in the taking of a Software Step exception. This erratum provides the ability for a virtual machine to deny use of the core to the Hypervisor. In some use models of virtualization, particularly a multitenant system with untrusted code running at EL1, such a capability by the virtual machine is extremely undesirable.

Workaround

No performance-acceptable workaround exists.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1467587
HCR_EL2.TOCU incorrectly applies during EL0 execution when HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

The trap control bit HCR_EL2.TOCU is incorrectly applied at EL0 when HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1. Under these conditions, IC IVAU and DC CVAU instructions at EL0 should not be exceptional, but they incorrectly trap to EL2 with an ESR_EL2 code of Trap (EC=0x18).

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing at EL0 with HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1.
2. The core executes an IC IVAU or DC CVAU instruction.

Implications

If the above conditions are met, then the IC IVAU or DC CVAU instruction is incorrectly trapped to EL2.

Workaround

This erratum can be avoided by clearing HCR_EL2.TOCU when entering HCR_EL2.(E2H,TGE)=(1,1) mode.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1533195
Accessing a memory location using mismatched shareability attributes might cause loss of coherency

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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 configurations with only a single core and without a DSU L3 cache. Such systems are defined as direct connect using the following RTL parameter values:

```
NUM_BIG_CORES: 1
L3_CACHE: FALSE
ACE: FALSE
PORTER_SAM: TRUE
ACP: FALSE
PERIPH_PORT: FALSE
ASYNC_BRIDGE: TRUE
```

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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1688567
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1688568
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1791580
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1800710
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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:

- Both stage 1 and stage 2 translations are enabled.
- Stage 1 page or block size is larger than stage 2 page or block size.
- MMU TC RAM entry has a transient single-bit ECC error.
- TLBI targets the translation in the MMU TC RAM entry containing the single-bit ECC error.
- The single-bit ECC error prevents the TLBI from removing the entry.
- 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 condition can be detected by ERR0STATUS[25:24] == 0b10, indicating a corrected error, and ERR0MISC0[3:0] == 0b0010, indicating the source to be the MMU TC RAM. If enabled, the fault handling interrupt is asserted when an error is recorded in the ERR0* error record, and software can check for this condition. Software should treat this condition as an uncontainable uncorrected error (i.e. as if ERR0STATUS[29,21:20] == 0b100).

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1850713 Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

If a load or store crosses a cache line (cache line size = 64 bytes) and a watchpoint address targets a location in the upper cache line, the Fault Address Register (FAR) or the External Debug Watchpoint Address Register (EDWAR) (if set up for Debug Halt) will contain an incorrect address.

Configurations Affected

This erratum affects all configurations.

Conditions

Incorrect address in FAR or EDWAR appears when the:

1. Watchpoint targets a double word (or less or more) at cache line address B.
2. Load or store targets accesses two cache lines: lower cache line A and upper cache line B. The cache line size is 64 bytes.

Implications

FAR contains the target address of load or store.

EDWAR contains the target address of load or store if enabled for Debug Halt.

Workaround

There is no hardware workaround.

The following software workaround can be applied:

If the Fault Address Register (FAR) or External Debug Watchpoint Address Register (EDWAR) does not match a watchpoint, software can attempt to identify a relevant watchpoint:

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

a) For A DC ZVA whose address is not aligned to DCZID_EL0.BS, by rounding the faulting address down to a cache line boundary (64 bytes) and attempting to match this against active watchpoints.

Note: Most software aligns addresses used by DC ZVA, and this case is expected to be rare in practice.

b) For all other loads and stores, by attempting to use the address of the next cache line boundary (64 bytes) and attempting to match this against active watchpoints.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1868343
The core might update ELR_ELn with an incorrect value when the core is stepping a conditional branch instruction located at the end of 32-byte boundary

Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

When the core executes a conditional branch instruction with software step or halt step, the core might write an incorrect address into ELR_ELn after the core completes stepping.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is stepping a conditional branch instruction located at the end of a 32-byte aligned block.
2. The conditional branch is resolved as not taken.

Implications

If the above conditions are met, the core might not write the correct instruction address (PC+4 of stepping instruction) into the ELR_ELn register after stepping is completed.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1, which delays instruction fetch after branch misprediction. This workaround will have a small impact on performance.

The workaround for this erratum is the same as the workaround for errata 1262606 and 1275112.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1923202
External debugger access to Debug registers might not work during Warm reset

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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:
   a. DBGBCR<n>_EL1 (n=0-5)
   b. DBGBVR<n>_EL1 (n=0-5)
   c. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1946160
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, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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, a memory ordering violation might happen.

Workaround

There is no workaround for this erratum for versions r0p0, r1p0, and r2p0.

This erratum can be avoided in versions r3p0, r3p1, r4p0, and r4p1 by inserting a DMB ST before acquire atomic instructions without release semantics. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers:

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

```
LDR x0,=0x3
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,=0x4
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,=0x5
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
```


Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1978083
Incorrect programming of PMBPTR_EL1 might result in a deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2356586
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2743102
The core might deadlock during powerdown sequence

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

Description

While powering down the *Processing Element* (PE), a correctable L2 tag ECC error might cause a deadlock in the power-down 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 power-down sequence as described in TRM.

Implications

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

Workaround

Add a DSB instruction before the ISB of the power-down code sequence specified in the TRM.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3023823
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, r2p0, r3p0, r3p1, r4p0 and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3324349
MSR PSTATE.SSBS to 0 is not fully self-synchronizing

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3696297
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, r2p0, r3p0, r3p1, r4p0 and r4p1. 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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Category B (rare)

1286807
Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation

Status

Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

If a virtual address for a cacheable mapping of a location is being accessed by a core while another core is remapping the virtual address to a new physical page using the recommended break-before-make sequence, then under very rare circumstances TLBI+DSB completes before a read using the translation being invalidated has been observed by other observers.

Configurations Affected

The erratum affects all multi-core configurations.

Conditions

1. Core A speculatively executes a load (LD2) ahead of an older load (LD1) to the same cacheable virtual address.
2. Core B marks the associated translation table entry invalid, followed by a DSB; TLBI; DSB sequence which generates a sync request.
3. LD2 returns its result using the original physical address (PA1) under specific narrow timing conditions before Core A has responded to the sync request.
4. Core B receives the response and updates the translation table entry to map a new physical address (PA2) followed by a DSB.
5. LD1 returns its result using PA2 on Core A and commits the result from LD2 using PA1 because the read-ordering violation is not detected.

Implications

If the above conditions are met under certain timing conditions, then this erratum might result in a read ordering violation.

Workaround

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

This erratum can be avoided by executing the TLB invalidate and DSB instructions a second time before modifying the translation table of a virtual page that is being accessed by an active process.

Note: For code sequences which have multiple TLB invalidate instructions followed by a single DSB, only the last TLB invalidate and DSB need to be repeated a second time.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1418040
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, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

- CNTPCT.
- CNTVCT.
- CNTFRQ.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1542419
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 r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

When the core executes an instruction that has been recently modified, the core might fetch a stale instruction, which violates the ordering of instruction fetches. This is due to the architecture changes involving prefetch-speculation-protection.

Configurations Affected

This erratum affects all configurations with

```
COHERENT_ICACHE
```

set to

```
TRUE
```

Conditions

1. An instruction A, residing at physical address X is modified by another instruction, A'.
2. After instruction A is modified and made visible to all other processors, another instruction, B, residing on physical address Y is modified by instruction B'.
3. The core fetches the instruction from physical address Y, and fetches instruction B'.
4. After the instruction fetch for instruction B', the core fetches the instruction from physical address X before executing a context synchronization instruction.

Implications

If the above conditions are met, then the core might execute a stale instruction (instruction A) instead of the up-to-date instruction (instruction A').

Workaround

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

The erratum affects software depending on prefetch-speculation-protection, instead of explicit synchronization, at any exception level. Privileged exception levels are expected to have a mechanism to workaround this erratum with either an inner-shareable TLBI followed by a DSB, or an ISB by the executing PE. The following workaround focuses on an operating system providing the workaround on behalf of EL0.

For software running at EL0, this erratum can be avoided by executing TLB inner-shareable invalidation operation followed by DSB between condition 1 and condition 2 by trapping IC IVAU instructions to EL3, whereby the trap handler executes the TLB inner-shareable invalidation and DSB operations. This is accomplished by setting up the following:

1. Trap EL0 accesses to CTR_EL0 by setting SCTLR_EL1.UCT to 0 and emulating accesses so that the DIC field appears as RES0 to EL0 software. Since one TLB inner-shareable invalidation is enough to avoid this erratum, the number of injected TLB invalidations should be minimized in the trap handler to mitigate the performance impact due to this workaround. This is accomplished by making the IminLine[3:0] field appear as 0b1111.
2. Trap all EL0 IC IVAU instructions to EL3 by using the following write sequence to several IMPLEMENTATION DEFINED registers:

```
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,=0xEE670D35
MSR S3_6_c15_c8_2,x0
LDR x0,= 0xFFFF0FFF
MSR S3_6_c15_c8_3,x0
LDR x0,=0x08000020007D
MSR S3_6_c15_c8_1,x0
ISB
```

   The above indicates an IMPLEMENTATION DEFINED exception to EL3 with the ESR_EL3.EC set to 0b11111 (0x1F).

3. In response to the trap to EL3 from EL0, the trap handler executes a TLB inner-shareable invalidation to an arbitrary address followed by a DSB.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Category C

901361
Failure to report or incorrect reporting of L2 data RAM ECC errors

Status

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

Description

For certain operation types, a data read from the L2 data RAMs caused by a cache line victimization might fail to report and log a RAS error if such data contains a single or double bit ECC error. In some cases, an error is reported, but the physical address recorded is incorrect.

Configurations Affected

This erratum affects all configurations.

Conditions

When performing a refill of the L2 cache on behalf of an instruction fetch, load, store, or table walk, a data read from the L2 data RAMs for the cache line being replaced encounters a single or double bit ECC error.

Implications

If this erratum occurs, either:

- The L2 data RAM ECC error is detected, but the error is not reported or logged in the CPU RAS registers.
- The error is reported and logged, but the wrong physical address is recorded.

Error recovery software will not be able to correctly determine the source of a data error.

Note that any required error propagation to consumers of the data from the L2 data RAMs in the form of poison occurs correctly.

Workaround

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

A partial workaround is possible. Setting CPUACTLR2_EL1[45] to 1 forces reporting and logging of all L2 data RAM ECC errors. However, the reported physical address might still be incorrect. Other recorded information, such as array, subarray, and index are correct. Setting CPUACTLR2_EL1[45] to 1 might have a small impact on performance.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

901865
Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock

Status

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

Description

Under certain conditions, an LDREX/STREX loop might continuously mispredict. If the speculative instruction path has a load or store to the same Physical Address (PA) as the exclusive monitor address, but with a different Virtual Address (VA), this might cause a repeatable loop where the cache line is lost, opening the exclusive monitor.

Configurations Affected

The erratum affects all configurations.

Conditions

1. The LDREX/STREX loop has a branch that is consistently mispredicted. This includes all Device memory code, which the branch predictor does not train.
2. There is a load or store outside of the loop that has the same PA as the exclusive monitor address, within a cache line. However, this load or store has a different VA, specifically VA[13:12] for 64KB L1 cache. The load or store makes a request to the L2 that snoops the L1, opening the exclusive monitor. The Arm architecture disallows a load or store inside an LDREX/STREX loop with VA aliasing to the exclusive monitor cache line.

Implications

If the above conditions are met, the core might livelock.

Workaround

This erratum is not expected to be encountered in real software. There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

902290
Persistent error response to transactions issued on behalf of Page descriptor Access bit and Dirty bit updates might livelock

Status

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

Description

A Page descriptor Access bit and Dirty bit update request by the MMU might not be successful if accessing the memory location encounters an uncorrectable error. In this case, the MMU might retry the request repeatedly instead of reporting an external abort.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core cache has the descriptor in SharedClean state.
2. A ReadUnique request is sent to the system to update the Access Flag bit or the Dirty bit of the descriptor.
3. The system responds with a Non-Data Error (NDErr).

Implications

If the above conditions occur in a persistent manner, the core might livelock.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

909055
Failure to record Level 1 data cache access event when using the SPE

Status

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

Description

The recording of the Level 1 data cache access event into the Statistical Profiling Extension (SPE) buffer is inaccurate. It is possible for a sampled micro-operation to incorrectly indicate that it did not access the data cache when it in fact did, and conversely it is possible for the sampled micro-operation to indicate that it did access the data cache when in fact it did not.

Configurations Affected

This erratum affects all configurations.

Conditions

When determining whether a sampled micro-operation accessed the Level 1 data cache, the result is incorrect unless there is an unrelated micro-operation which executes simultaneously with the sampled micro-operation and has the same cache access behavior.

Implications

If this erratum occurs, the contents of the SPE buffer is inaccurate with regards to the E[2] bit.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

930017
Failure to sign-extend instruction virtual address when using the SPE

Status

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

Description

The instruction virtual address bits[55:49] included in the Statistical Profiling Extension (SPE) record are not properly sign-extended. Bits[55:49] are always set to 0.

Configurations Affected

This erratum affects all configurations.

Conditions

The instruction virtual address bits[55:48] are all equal to 1.

Implications

When the instruction virtual address bits[55:48] are all equal to 1, the value written into the Statistical Profiling Extension buffer for bits[55:49] is not correct.

Workaround

Software can examine instruction virtual address bit[48] to determine the proper value for all bits in the range [55:48]. If bit[48] is equal to 0, then all bits in the range should be read as 0. If bit[48] is equal to 1, then all bits in the range should be read as 1.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

933092
Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort

Status

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

Description

Under certain conditions, an external error or poison flagged on a critical beat response for a linefill might get dropped. As a result, a load that forwards data from that critical beat, might return data without signaling an abort.

Configurations Affected

The erratum affects all configurations.

Conditions

1. The Processing Element (PE) executes one or more load instructions that miss in both the L1 and L2 caches, and initiates a linefill request.
2. The system returns the critical beat tagged with either poisoned data or an external error indication.
3. The critical beat is returned to the Load/Store (LS) unit of the PE, and no more data beats are returned to the LS for three or more cycles.
4. Load instructions that can complete by getting their data from the critical beat, are sent down the pipeline so that they pick up data three or more cycles after it was returned to the LS.

Implications

If the above conditions are met, the PE might consume poisoned data or data tagged with an error, without signaling an abort.

Workaround

A workaround is not expected to be necessary in most cases, as this erratum will only cause a negligible increase in the failure in time (FIT) rate.

If a workaround is required, set CPUACTLR2_EL1[43] to 1. This prevents critical beat forwarding, and ensures that the load will abort in case the system reports an error. Setting CPUACTLR2_EL1[43] to 1 might have a small impact on performance.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

933779
DBGDTRTX register fails to hold value through Warm reset

Status

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

Description

The DBGDTRTX register is architecturally required to hold value through a Warm reset. Because of this erratum, a reset connection error has it resetting the value on a Warm reset instead.

Configurations Affected

This erratum affects all configurations.

Conditions

DBGDTRTX is holding a value other than the initial reset value and a Warm reset occurs.

Implications

If this erratum occurs, the content of the DBGDTRTX register is inaccurate.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

934968
DCPSx instruction with SCTLR_EL1.IESB = 1 while in debug state might not execute correctly

Status

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

Description

A DCPS1, DCPS2, or DCPS3 instruction that is executed in debug state while SCTLR_EL1.IESB is set to 1 might result in incorrect execution of the instruction.

Configurations Affected

This erratum affects all configurations.

Conditions

1. SCTLR_EL1.IESB is set to 1.
2. The core is in debug state.
3. The core executes a DCPS1, DCPS2, or DCPS3 instruction.

Implications

If the above conditions are met, then this erratum might result in deadlock, data corruption, or produce other undesirable effects. However, this erratum will not result in violation of access controls, for example, this erratum will not result in the core making accesses to Secure memory from Non-secure mode.

Workaround

The erratum can be avoided by clearing SCTLR_EL1.IESB before executing a DCPSx instruction in debug state. If the core is in a state where SCTLR_EL1 writes are trapped, then up to three write attempts might be required where each attempt might be trapped to a higher Exception level.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

937437
An SPE buffer full event might clear PMBSR_EL1.DL and PMBSR_EL1.EA

Status

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

Description

When there is not sufficient space in the Statistical Profiling Extension (SPE) profile buffer to write another record, a buffer management event is generated. This event should not alter the value of PMBSR_EL1.DL and PMBSR_EL1.EA. However, under certain conditions the buffer full event might clear these bits.

Configurations Affected

This erratum affects all configurations.

Conditions

There are two cases in which the erratum might occur.

Case A:

1. An explicit write of PMBSR_EL1 using an MSR instruction has set either PMBSR_EL1.DL or PMBSR_EL1.EA.
2. PMBSR_EL1.S is not set.
3. A buffer full event occurs.

Case B:

1. An External abort occurs on a write to the SPE profile buffer, but the indication of that abort has not yet been received by the core.
2. A subsequent write to the profile buffer is initiated, which will cause a buffer full event.
3. The indication of the External abort is received in a small window of time immediately before the buffer full event.

If the above conditions are met for either of the two cases, then the buffer full event might cause PMBSR_EL1.DL and PMBSR_EL1.EA to be cleared.

Note that Case A is not possible if software always initializes the value of PMBSR_EL1.DL and PMBSR_EL1.EA to zero, and always restores the value of PMBSR_EL1.DL, PMBSR_EL1.EA, and PMBSR_EL1.S to the value previously held in PMBSR_EL1 on a save and restore operation.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Implications

If the above conditions are met, PMBSR_EL1.DL and PMBSR_EL1.EA will be cleared. This means that there will no longer be a record that an External abort caused the profile buffer to become corrupted.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

941868
Deferred errors might cause silent data corruption following a hardware update of Access and Dirty bits in a translation table entry

Status

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

Description

An L2 linefill caused by a hardware update to the Access and Dirty bits in the translation tables might cause silent data corruption if the data received from the system contains an external error or poison indication.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Error correction is disabled with ERR0CTLR.ED set to 0.
2. A memory access causes a hardware update of the Access and/or Dirty flags of the corresponding translation table entry.
3. The hardware update causes the L2 cache to initiate a linefill operation because of a cache miss or hit to SharedClean state.
4. The data returned from the system contains a deferred error indicated by poison, a data error, or non-data error responses in data other than the doubleword containing the Access and Dirty flags.

Implications

If the above conditions are met, the translation table Access and Dirty bits might not be updated if the cache line containing the translation table entry contains a deferrable data error.

Workaround

If the system supports far atomic accesses to cacheable memory, then setting CPUACTLR2_EL1[41] to 1 forces the L2 cache to perform hardware updates of the Access and Dirty flags as far atomics.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

944783
Address breakpoint might cause a deadlock with certain AArch32 T32 code sequences

Status

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

Description

If an Address breakpoint is set on a T32 instruction, then under certain conditions the core might stop executing a few instructions before the Breakpoint exception should occur.

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. The breakpoint is not quadword aligned.
4. The cache line contains at least two 32-bit instructions, of which at least one must be after the breakpoint.

Implications

If the above conditions are met, the processor might deadlock.

Workaround

Any interrupt will break the processor out of the deadlock state. The deadlock can be avoided by forcing the page containing the T32 instruction to be Non-cacheable.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

961111
L2 might report multiple RAS errors for the same prefetch request

Status

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

Description

If a hardware or software prefetch targeting the L2 encounters a tag ECC error and hazards against an outstanding request for the same cache line, then multiple RAS errors might be reported.

Configurations Affected

This erratum affects all configurations.

Conditions

A hardware or software prefetch operation that hazards against an outstanding read request for the same cache line and detects a tag ECC error might allow the internal signals for RAS errors to remain asserted, leading to multiple reported errors for the same operation.

Implications

If this erratum occurs, then the ERR0STATUS.OF bit might be set when only one error has actually occurred.

Workaround

No workaround is required for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

964384
Stuck-at-fault in L1 instruction cache data array might cause deadlock with certain AArch32 T32 code sequences

Status

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

Description

Detected parity errors in the L1 instruction cache data array will trigger a line fill request to repeat the instructions. In certain scenarios, the returned data is not used directly but is first stored in the cache. If a stuck-at-fault is present, then the core will continuously request the same line fill and no further instructions will be executed.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The cache line contains a 32-bit instruction starting at odd halfword alignment.
3. The upper 5 bits of the second halfword of this 32-bit instruction must be 0b11101, 0b11110, or 0b11111.
4. A stuck-at-fault must exist in the second halfword of the instruction.

Implications

If the above conditions are met, the processor might deadlock.

Workaround

Any interrupt will break the processor out of the deadlock state. The stuck-at-fault can be bypassed by forcing the page containing the T32 instruction to be Non-cacheable.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

978245
Executing unallocated encoding in conversion between floating-point and integer instruction class does not generate Undefined Instruction exception

Status

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

Description

If the core executes the following unallocated encoding in the A64 conversion between floating-point and integer instruction class, where sf = x, S = 0, type = 11, rmode = 01, opcode = 11x, then instead of taking an Undefined Instruction exception, the core incorrectly executes this unallocated encoding as a vector half-precision "FABD <Vd>.<T>, <Vn>.<T>, <Vm>.<T>" instruction.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing in AArch64 state.
2. An unallocated encoding in the conversion between floating-point and integer instruction class, where sf = x, S = 0, type = 11, rmode = 01, opcode = 11x, is executed.

Implications

If the above conditions are met, the core does not take an Undefined Instruction exception.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

986709
MRS to DBGDTR_EL0 might cause EDSCR.RXfull bit to clear incorrectly

Status

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

Description

An MRS to DBGDTR_EL0 which is speculatively executed might cause the EDSCR.RXfull bit to incorrectly clear before the data is read from DBGDTR_EL0.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The EDSCR.RXfull bit is set to 1.
2. An MRS to DBGDTR_EL0 is speculatively executed.

Implications

If the above conditions are met, then the EDSCR.RXfull bit might be cleared before the data is read from DBGDTR_EL0. This might cause:

- The core to not receive all data when an external debugger sees the RXfull bit cleared and, as a result, sends new data before the core receives the old data.
- Earlier Instructions in the instruction stream to see the RXfull bit cleared out of program order.

Workaround

This erratum can be avoided by inserting an ISB instruction before the MRS to DBGDTR_EL0, because the ISB can prevent the MRS from being speculatively executed.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

988575
Unaligned cache line split load to NC or Device memory, tagged with poison or external error on its first half, might cause data corruption

Status

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

Description

Under certain conditions, an external error or poison flagged on the first half of an unaligned load to Non-Cacheable (NC) or Device memory that crosses a cache line boundary, might get dropped. As a result, the unaligned NC or Device load can return data without signaling an abort.

Configurations Affected

The erratum affects all configurations.

Conditions

1. The Processing Element (PE) executes an unaligned load to NC or Device memory that crosses a cache line boundary.
2. The system returns data for the first line tagged with either poisoned data or an external error indication.
3. The second half of the cache line split is not tagged with either poisoned data or an external error.
4. The load, on receiving data for the first half, executes such that the second half of the unaligned load gets squashed by either an older load or some other high priority requestor and will be replayed.

Implications

If the above conditions are met, the PE might consume poisoned data or data tagged with an error, without signaling an abort.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1051464
CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0. Fixed in r2p0.

Description

A CTI trigger arriving on the same cycle the core debug block receives the **PREADYCD** from an outstanding CTI trigger transaction might result in the arriving CTI trigger to be lost. This erratum occurs only on CTI trigger events from the core to the external debug block.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Multiple CTI triggers are generated in the core going to the external debug block in close time proximity.
2. A CTI trigger event arrives in the same cycle the core debug block receives **PREADYCD** from an outstanding CTI trigger transaction.

Implications

If the above conditions are met, then the arriving CTI trigger might be lost and the system does not receive a trigger event. If the trigger event is from the same CTI source, then no issues will be seen as CTI triggers are allowed to be merged. If the trigger is from a different CTI source, then the debug system might not behave in an optimal fashion although debug operations can continue.

Workaround

No workaround is required for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1057923
Extra instruction might be executed during Halting Step when stepping WFI, WFE, and some self-synchronizing system register writes

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0. Fixed in r2p0.

Description

During Halting Step, execution of a small set of instructions in the Active-not-pending state can result in the execution of that instruction and the next instruction before returning control to the debugger by entering debug state. That is, instead of a single instruction executed between returns to the debugger, two instructions are executed. The set of instructions that can cause the stepping of an extra instruction is WFE, WFI, and some self-synchronizing system register writes.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Core is in Halting Step mode.
2. The instruction being stepped is either a WFE, a WFI, or some self-synchronizing system register writes.

Implications

If the above conditions are met, then two instructions will be stepped when a single step is expected, causing a potential DLR_EL0 mismatch by software. However, the instructions still execute in the correct order and function correctly.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1069401
Debug APB accesses to the ELA RAM might return incorrect data

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0. Fixed in r2p0.

Description

Debug APB registers support direct write and read accesses to the ELA RAM, using the following registers:

- RAM Write Address Register (RWAR).
- RAM Write Data Register (RWDR).
- RAM Read Address Register (RRAR).
- RAM Read Data Register (RRDR).

After writing the ELA RAM using the RWAR/RWDR registers, a subsequent read access using the RRAR/RRDR registers might return old data from the RRDR register. The ELA RAM read operation is dropped and the previous contents of the RRDR register are returned instead of the contents associated with the current operation.

Configurations Affected

This erratum affects all configurations with ELA set to TRUE.

Conditions

1. Write RWAR register with target index of ELA RAM.
2. Write RWDR register with write data to trigger the ELA RAM write access.
3. Write RRAR register with target index of ELA RAM to trigger the ELA RAM read access.
4. Read RRDR register to return the data.

Implications

If the above conditions are met, then old data is returned from the RRDR register because the write to the RRAR register to trigger the ELA RAM read access is dropped.

Workaround

This erratum can be avoided by inserting two writes to the ELA Lock Access Register (LAR) before writing the RRAR register.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1096402
Exception packet for return stack match might return incorrect [E1:E0] field

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

When an abort or trap is taken at the target of an indirect branch matching the return stack value in the core ETM, an Exception packet might be generated with the 2-bit field [E1:E0] = 0b10, which implies an Address element before the Exception element. When there is a trace return stack match, an Address element should not be generated before the Exception element. With [E1:E0] = 0b10, the external Trace Analyzer might read the trace packet sequence to expect an Address element output before the Exception element and not complete the stack pop, which is incorrect. The correct value in the [E1:E0] field in the Exception packet for this case, should be 0b01.

Configurations Affected

This erratum affects all configurations.

Conditions

1. ETM is enabled.
2. TRCCONFIGR.RS = 1, which indicates the return stack is enabled.
3. Abort or trap is taken at the target of an indirect branch matching the return stack.

Implications

If the above conditions are met, then the external Trace Analyzer does not pop on the return stack match, causing it to go out of sync with the core ETM.

Workaround

If tracing only EL0, then no workaround is required.
Otherwise, setting TRCCONFIGR.RS = 0 to disable return stack is the workaround.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1109624
Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

Under certain conditions, an LDREX/STREX loop might continuously mispredict. If the speculative instruction path has a load or store to the same Physical Address (PA) as the exclusive monitor address, but a different Virtual Address (VA), then this might cause a repeatable loop where the cache line is lost, opening the exclusive monitor.

Configurations Affected

The erratum affects all configurations.

Conditions

1. The LDREX/STREX loop has a branch that is consistently mispredicted. This includes all Device memory code, which the branch predictor does not train.
2. There is a load or store outside of the loop that has the same PA as the exclusive monitor address, within a cache line. However, this load or store has a different VA, specifically VA[13:12] for 64KB L1 cache. The load or store makes a request to the L2 that snoops the L1, opening the exclusive monitor. The Arm architecture disallows a load or store inside an LDREX/STREX loop with VA aliasing to the exclusive monitor cache line.
3. The LDREX-STREX loop also contains an ISB instruction.

Implications

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

Workaround

This erratum is not expected to be encountered in real software. There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1119735
16-bit T32 instruction close to breakpoint location might cause early breakpoint exception

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

If an address breakpoint is set on the instruction following a 16-bit T32 instruction, then under certain conditions the core might trigger the breakpoint on that 16-bit T32 instruction. This can happen if there is a parity error on the 16-bit T32 instruction before the breakpoint, or if the 16-bit T32 instruction has different cacheability than prior instructions.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing an AArch32 T32 code sequence.
2. A breakpoint is set on the instruction following a 16-bit T32 instruction.
3. One of the following conditions is true:
    - The breakpoint instruction follows a 16-bit T32 instruction containing a parity error.
    - The breakpoint instruction and the prior 16-bit T32 instruction both belong to a cache line that has different cacheability than the previous cache line.

Implications

If the above conditions are met, then the breakpoint might be triggered on the preceding 16-bit T32 instruction.

Workaround

There is no workaround for this erratum. This situation can be detected by reading the contents of the appropriate ELR_ELx register after the breakpoint exception has been taken.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1126105
Read from L1 instruction cache data array using RAMINDEX operation might return data from the wrong location

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

The RAMINDEX operation can be used to read the L1 instruction cache data array contents using the Index and Way fields. Because of this erratum, bits of the Index field of the RAMINDEX operation are swapped and Index {Index[13:6],Index[4:3],Index[5]} is used instead of Index[13:3].

Configurations Affected

This erratum affects all configurations.

Conditions

A RAMINDEX operation is performed targeting the L1 instruction cache data array.

Implications

Data read from the RAMINDEX operation targeting the L1 instruction cache data array might not come from the specified Index field of the RAMINDEX operation.

Workaround

The Index field for the RAMINDEX operation can be adjusted appropriately to access the desired L1 instruction cache data array entry.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1144394
Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

During software step, execution of some load instructions in the Active-not-pending state might result in the execution of that instruction and the next instruction before returning control to debugger software by taking a software step exception, instead of returning after a single instruction executed.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in software step mode.
2. The instruction being stepped is a load instruction that loads two or more destination registers.
3. Snoop invalidation of a cache line referenced by the load occurs during its execution, or an ECC error response occurs on the load.

Implications

If the above conditions are met, then two instructions can be stepped when a single step is expected, causing a potential ELR_ELx mismatch by software. However, the instructions still execute in the correct order and function correctly.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1192279 IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

If the interconnect does not support atomic memory operations, then instructions which try to perform these to Non-cacheable or Device memory take an IMPLEMENTATION DEFINED fault with Data Fault Status Code of ESR_ELx.DFSC = 0b110101. If the PE is executing at EL0 or EL1, Stage 2 translation is enabled, and HCR_EL2.CD forces the final memory type to be Non-Cacheable, then this fault is not routed to EL2.

Configurations Affected

The erratum affects all configurations.

Conditions

1. The interconnect does not support atomic operations.
2. The PE is executing at EL0 or EL1.
3. There is an atomic instruction to memory which is mapped as Non-cacheable because Stage 2 translation is enabled and HCR_EL2.CD is set.

Implications

If the above conditions are met, then the IMPLEMENTATION DEFINED fault with Data Fault Status Code of ESR_ELx.DFSC = 0b110101 is not routed to EL2.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1194748
The ERXADDR_EL1 register might report an incorrect physical address for an L1 data tag RAM single-bit correctable ECC error

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

If a load, store, prefetch, or snoop request encounters a single-bit correctable ECC error in the L1 data cache tag RAM, then the physical address captured in the ERXADDR_EL1 register might be incorrect, even if the ERXSTATUS_EL1.AV bit is set to 1, indicating a valid address.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An L1 data cache tag RAM lookup on behalf of a load, store, prefetch, or snoop request encounters a single-bit correctable ECC error.
2. The address of the line that has the ECC error is not the address that is being looked up in the L1 data cache tag RAM.

Implications

If this erratum occurs, then the ERXSTATUS_EL1.AV bit is set to 1, but the address captured in the ERXADDR_EL1 register is not the correct physical address of the line that had the single-bit correctable ECC error.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1194749
ERR0MISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

If a parity error is detected in the L1 instruction cache data array, then the error location might not be computed correctly. This results in incorrect BANK and SUBBANK information in the ERR0MISC0 register.

Configurations Affected

This erratum affects all configurations.

Conditions

A parity error is detected in the L1 instruction cache data array.

Implications

If the above conditions are met, then the BANK and SUBBANK fields of the ERR0MISC0 register might have incorrect information. This does not impact other fields in the ERR0MISC0 register that apply to the L1 instruction cache.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1214504
Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

An IMPLEMENTATION DEFINED instruction that reads the contents of the L1 data TLB after a context switch might report an incorrect value of the valid bit for the corresponding TLB entry.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An instruction to perform a direct access to the L1 data TLB is present in program order before a context switch event.
2. The read of the L1 data TLB contents as part of the direct access instruction occurs after the context switch.

Implications

If the above conditions are met, then an incorrect value might be reported for the valid bit of the L1 data TLB entry being accessed directly.

Workaround

This erratum can be avoided by inserting a DSB after every instruction that accesses the L1 data TLB directly.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1227053
Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

Writes to contiguous bytes might be coalesced into one streaming write of 64 bytes. If such writes are performed to memory mapped Non-shareable and write-back, then two streaming writes to the same physical address might be performed in the wrong order.

Configurations Affected

This erratum affects configurations without a DSU L3 cache and snoop filter. Such systems are defined as direct connect using the following RTL parameter values:

- L3_CACHE: FALSE.
- ACE: FALSE.
- PORTER_SAM: TRUE.
- ACP: FALSE.
- PERIPH_PORT: FALSE.
- ASYNC_BRIDGE: TRUE.

Conditions

Write stream operations to memory mapped Non-shareable and write-back, or shareable and write-back with the *BROADCASTOUTER *pin deasserted can allocate the L2 cache without issuing a request on the CHI interface. This creates the possibility of two concurrent pending WriteNoSnpFull transactions of the same cache line on CHI without the proper sequencing to guarantee their order of performance.

Implications

If the above conditions are met, then the coalesced writes might be performed in the wrong order as determined by the sequential execution model.

Workaround

This erratum can be avoided by mapping all write-back memory as Inner or Outer Shareable.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1227629
ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r3p0.

Description

The ERR0STATUS.SERR field is updated incorrectly for Error responses from slave and Deferred errors from slave not supported at master. Error responses from the interconnect for copyback transactions should record ERR0STATUS.SERR = 0x12. Because of this erratum, they incorrectly record 0x18. Undeferrable data errors received from the interconnect should record ERR0STATUS.SERR = 0x15. Because of this erratum, they incorrectly record 0x12.

Configurations Affected

This erratum affects all configurations.

Conditions

The core issues a copyback transaction (WriteBackFull, WriteEvictFull, Evict, or WriteNoSnpFull) which then receives an error response.

Implications

If the above condition is met, then the ERR0STATUS.SERR field is incorrect and software handling these errors reports the wrong class of error.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1244984
Illegal return event might corrupt PSTATE.UA0

Status

Fault Type: Programmer Category C
Fault Status: Present on r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

An illegal return event from AArch64 state erroneously updates PSTATE.UAO from the saved process state bit[23] when the saved process state stipulates an intended return to AArch32. The correct behavior is to leave PSTATE.UAO unchanged.

Configurations Affected

This erratum affects all configurations.

Conditions

- An illegal return event from AArch64 state occurs. This involves at least one of the following, where the saved process state stipulates return to a mode or state that is illegal:
  - Execution of an ERET instruction.
  - Execution of a DRPS instruction in Debug state.
  - Exit from Debug state.
- The saved process state specifies the AArch32 target execution state. The saved process state bit, M[4], is 1.

Implications

PSTATE.UAO might be corrupted.
This corrupted value is saved in SPSR_ELx on taking an Illegal Execution state exception or an asynchronous exception immediately after the illegal return event. The corrupted PSTATE.UAO has no impact on instruction execution until returning from the Illegal Execution state exception handler.

Workaround

No workaround is required for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1256788
Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

During Halting Step, execution of some load instructions in the Active-not-pending state might result in the execution of that instruction and the next instruction before returning control to the debugger by entering Debug state, instead of returning after a single instruction executed.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in Halting Step mode.
2. The instruction being stepped is a load instruction that loads two or more destination registers.
3. Snoop invalidation of a cache line referenced by the load occurs during its execution, or an ECC error response occurs on the load.

Implications

If the above conditions are met, then two instructions can be stepped when a single step is expected, potentially resulting in unexpected DLR_EL0 and DSPSR_EL0 values upon entry to Debug state. However, the instructions still execute in the correct order and function correctly.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1264383
Write-Back load after two Device-nG* stores to the same physical address might get invalid data

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, and r3p0. Fixed in r3p1.

Description

In certain circumstances, a load to Write-Back memory might get a logical OR of two Device-nG* stores to the same physical address. This does not happen with proper break-before-make page remapping, and only happens with two virtual addresses mapped to the same physical address and mismatched attributes. A data cache maintenance operation to this physical address between the stores and load to guarantee coherency also prevents this erratum. The load page translation needs to replace the store translation in the L1 data TLB, requiring accesses to 47 other pages in between.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Two stores to physical address A with Device-nG* memory attribute occur.
2. Load/store accesses to 47 or more pages occur.
3. A load to physical address A with Write-Back memory attribute occurs.

Implications

If the above conditions are met, then under specific microarchitectural conditions, the load returns data that is a logical OR of the two or more stores.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1346756
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, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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 will execute locally and broadcast 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1349291
Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

When an Uncontainable (UC) SError is reported or deferred by the core, it might be incorrectly logged as an Unrecoverable (UEU) SError. This is an inappropriate categorization downgrade which might allow for silent error propagation.

Configurations Affected

This erratum affects all configurations.

Conditions

1. An Uncontainable (UC) SError occurs in the system.
2. The Uncontainable (UC) SError is reported or deferred.

Implications

If the above conditions are met, then the ESR_ELx.AET or DISR_EL1.AET field might log the Uncontainable (UC) SError as an Unrecoverable (UEU) SError.

Workaround

This erratum can be mitigated by treating all SErrors reported with type Unrecoverable (UEU) as type Uncontainable (UC).

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1356341 L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

The L1D_CACHE access related PMU events 0x4, 0x40, and 0x41 and the L1D_TLB access related PMU events 0x25, 0x4E, and 0x4F are incorrectly counting non-memory read/write operations that must be excluded. Software prefetch instructions are counted as read accesses and all other instructions are counted as write accesses.

Configurations Affected

This erratum affects all configurations.

Conditions

A software prefetch (PRFM) instruction or one of the following non-memory write operations is issued to the Load/Store Unit:

- A barrier (DMB, DSB, ESB, or PSB).
- A TLB Maintenance Operation (TMO).
- A Cache Maintenance Operation (CMO).
- An Address Translation operation (AT).
- A debug RAM read operation.

Implications

If any of the non-memory read/write operations listed above are issued to the Load/Store Unit, then the PMU counts for events L1D_CACHE (0x4), L1D_CACHE_RD (0x40), L1D_CACHE_WR (0x41) or L1D_TLB (0x25), L1D_TLB_RD (0x4E), and L1D_TLB_WR (0x4F) are incremented incorrectly.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1395332
Read from PMCCNTR in AArch32 might return corrupted data

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

When PMCCNTR is configured to count core clock cycles, the result of a read from the PMCCNTR system register in AArch32 state might be corrupted. This corruption is predictable and occurs when the clock cycle count rolls over into the upper 32 bits of the register. For example, if PMCCNTR=0xFFFF_FFFF and a read is executed around the time the clock cycle count is incremented, then the value returned might be 0x1_FFFF_FFFF rather than 0x1_0000_0000.

Configurations Affected

This erratum affects all configurations.

Conditions

1. PMCCNTR is configured to count core clock cycles.
2. The lower 32 bits of PMCCNTR contains a value close to 0xFFFF_FFFF.
3. A read from PMCCNTR is performed in AArch32.

Implications

If the above conditions are met, then the read from the PMCCNTR register might return corrupted data.

Workaround

This erratum is not expected to require a workaround.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1406411
MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

An MSR DSPSR_EL0 instruction that is executed in debug state and alters the Debug Saved Program Status Register, might fail to update PSTATE.{N,Z,C,V,GE} values on exit from debug state. This erratum applies to both AArch32 (MCR DSPSR) and AArch64 (MSR DSPSR_EL0) operation.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is in debug state.
2. The core executes an MSR instruction to alter the Debug Saved Program Status Register.
3. The core exits debug state.
4. The core might expose the incorrect PSTATE through execution of a conditional instruction or a read of PSTATE.{N,Z,C,V,GE} state.

Implications

If the above conditions are met, then this erratum might result in data corruption, incorrect program flow, or produce other undesirable effects. However, this erratum will not result in violation of access controls, for example, this erratum will not result in the core making accesses to Secure memory from Non-secure mode.

Workaround

The erratum can be avoided by setting CPUACTLR_EL1[45] to 1 prior to exiting from debug state. Power consumption in the core will be higher when CPUACTLR_EL1[45] is 1, as this prevents dynamic clock gating within sections of the core.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1408724
Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

The errant behavior described in this erratum pertains solely to ETM reporting information, and strictly in close vicinity of ETM reporting of an indirect branch with a malformed branch target address (a programming error).

Information recorded in the ETM trace buffer for branch instructions includes the Virtual Address (VA) of the branch target. An indirect branch has a malformed branch target address when either the lowermost bits of the target address stipulate a misaligned instruction address, or the uppermost bits are non-canonical. Execution of an indirect branch with a malformed target address results in an Instruction Abort. ETM trace information correctly reports the malformed target address for the branch execution, and also correctly reports exception information for the Instruction Abort.
However, under rare circumstances, a few branches immediately preceding the indirect branch with malformed target address can incorrectly include the upper and lower portions of the malformed target address in the ETM trace information for the target of these earlier branches. Only the upper and lower portions of the branch target VAs are potentially mis-reported in the ETM trace information.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. ETM is enabled.
2. An indirect branch with a malformed branch target address is executed and traced.
3. Branch instructions immediately preceding the indirect branch with malformed target address are executed and traced.

Implications

If the above conditions are met, then within a tightly constrained window the branches immediately preceding the indirect branch with malformed target address might record partially corrupted target addresses in the ETM trace buffer.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Workaround

No workaround is required. The programming error should be evident to users from the ETM trace information pertaining to the indirect branch with a malformed branch target address and trace information from its resultant Instruction Abort.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1415323
Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

If a core detects a false miss due to a single bit L1 tag RAM ECC error when executing a younger load instruction that bypassed another load or barrier and completed by forwarding data from a prior store instruction, then an ordering violation might occur in the presence of a snoop request.

Configurations Affected

The erratum affects all multi-core configurations with CORE_CACHE_PROTECTION= 1.

Conditions

1. Core A has a cache line X resident in the L1 data cache with write permissions, and has one or more stores in flight.
2. Core A performs a load (LD1) out-of-order for line X, bypassing another load or a barrier. The load encounters a tag single-bit ECC error, which makes the line appear as a miss, it allocates a miss request buffer requesting the line from L2.
3. LD1 is able to complete by forwarding data from an older store.
4. The older store drains and updates the L1 data cache.
5. Core B sends a snoop for line X and the snoop is ordered ahead of the miss request from LD1.
6. Core B performs a store to the line X.
7. Core A then receives the line X on behalf of its read request from LD1 and allocates the line.
8. Core A does not detect an ordering violation for the following:
    - An older load LD2 now observes this newer store, or
    - LD1 bypassed a load with acquire or barrier and is now required to observe the newer store.

Implications

If the above conditions are met, then under specific microarchitectural timing conditions, there might be an ordering violation, such as a read after read violation.

This has been graded as Programmer Category C because Arm expects this erratum to have a negligible impact over the undetected ECC failure rate in real systems. The reason for this categorization is that the issue only occurs during a very short window in time when using a highly implausible code sequence involving racing writes by multiple different cores.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1430754
Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

If a write to the External Debug Registers occurs such that it activates an address breakpoint, then under certain conditions the core might stop executing a few instructions before the breakpoint exception should occur.

Configurations Affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The breakpoint is set on a cacheable line.
3. The breakpoint is set on a cache line that starts with the final 16 bits of a 32-bit instruction.
4. There is a stuck-at-fault in the L1 instruction data array near the breakpoint location.
5. The breakpoint is activated using the External Debug Registers while the core is fetching.

Implications

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

Workaround

Any interrupt will break the core out of the deadlock state.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1487185
Waypoints from previous session might cause single-shot comparator match when trace enabled

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1490853 TRCIDR3.CCITMIN value is incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1514034
Error Synchronization Barrier (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, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

If a Virtual SError is pending and masked at the current Exception level when an ESB instruction is executed, then the VDISR_EL2 update occurs properly but in some cases 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 is required by the Arm architecture. 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 is not expected to be required. This is because existing software only executes ESB instructions at EL2 and above. If your software executes ESB instructions at EL1 with the conditions described above, then contact Arm support for more details.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1523502
CPUECTLR_EL1 controls for the MMU have no affect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

The CPUECTLR_EL1 register contains IMPLEMENTATION DEFINED configuration and control options for the MMU. The MMU bits affected by this erratum are CPUECTLR_EL1[54:46]. Any changes to these values have no affect on the functionality or performance.

Configurations Affected

This erratum affects all configurations.

Conditions

Software updates to modify MMU control bits CPUECTLR_EL1[54:46] from reset values have no effect.

Implications

Software attempts to change the functionality or performance of the core by changing reset values of CPUECTLR_EL1[54:46] have no affect. The value is updated in the register correctly, such that any subsequent read of the CPUECTLR_EL1 register will return the expected data, however, the modifications have no affect on the behavior of the core.

Workaround

There is no workaround.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1627784
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, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1655746
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, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1662732
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 r3p0, r3p1, and r4p0. Fixed in r4p1.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1694299
Instruction sampling bias exists in SPE implementation

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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

There is no workaround.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1697035
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, r1p0, r2p0, r3p0, r3p1 and r4p0. Fixed in r4p1.

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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1779123
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1788066
Possible loss of CTI event

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1788068
Loss of CTI events during warm reset

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1814889
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1857203
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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1857622
Uncorrectable tag errors in L2 cache might cause deadlock

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1874565 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, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1880110
Noncompliance with prioritization of Exception Catch debug events

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. 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)

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1899209
Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1899433
PFG duplicate reported faults through a Warm reset

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1912195
SPE events for "Other" operation type records might be captured incorrectly

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

Description

When the Statistical Profiling Extension (SPE) samples the swap micro-operation of a Compare And Swap (CAS) instruction, such operation is categorized as an "Other" operation type.

However, SPE record events [10:8] and [5:2], which are set only for Load/Store operation types, might be set and captured incorrectly.

Configurations Affected

This erratum affects all configurations.

Conditions

SPE is enabled and the following condition is met :

1. The sampling mechanism samples the swap micro-operation of a CAS.

Implications

The SPE record for a CAS sample might have some unexpected events set.

Workaround

There is no workaround.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1913776
L2 data RAM may fail to report corrected ECC errors

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. Fixed in r4p1.

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

This erratum affects all configurations.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

1930283
The PE might deadlock if Pseudofault Injection is enabled in Debug State

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2001418
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, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2001723
Incorrect timestamp value reported in SPE records when timestamp capture is enabled

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2019409
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, r0p1, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2052428
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2110726
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, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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

The following debug component shares the offset alias with the EDITR register when the PE is configured with 20-PMUs:

- PMU - PMEVCNTR16[63:32] - Event Counter 16

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2141647
A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. 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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2153915
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, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2227007
PMU L1D_CACHE_REFILL_OUTER is inaccurate

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2238117
Reads of DISR_EL1 incorrectly return 0s while in Debug State

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2239143
DRPS instruction is not treated as UNDEFINED at EL0 in Debug state

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2263697
L1 Data poison is not cleared by a store

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2307838 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, r2p0, r3p0, r3p1, r4p0, and r4p1. Open.

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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2391683
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2486423
L1D_TLB access related PMU event increments more than once per memory access

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, and r3p1. Fixed in r4p0.

Description

The L1D_TLB access related PMU event 0x25 increments more than once per memory access due to TLB miss and refill conditions. This might lead to inconsistencies between other TLB related events, such as, L1D_TLB_REFILL PMU event 0x5 or attributable L1 data TLB miss rate.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. MMU is enabled.
2. Memory accesses result in a significant number of misses to the L1 data TLB.

Implications

Memory accesses which result in a significant number of L1 data TLB misses might increment L1D_TLB PMU event 0x25 more than expected exposing inconsistencies with other related L1 data TLB events.

Workaround

There is no workaround for this erratum.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2729172
Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2816904
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2910961 L2D_CACHE_WB_CLEAN overcounts

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1. r4p0 and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3605051
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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)

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3607350
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3633468
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, r2p0, r3p0, r3p1, r4p0, and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3700183
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, r2p0, r3p0, r3p1, r4p0 and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3705920
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, r2p0, r3p0, r3p1, r4p0 and r4p1. 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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Proprietary notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

 Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm's trademark usage guidelines at **https://www.arm.com/company/policies/trademarks**. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(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.

Copyright © 2024 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential