



## Arm Cortex-X1 (MP077)

### Software Developer Errata Notice

Date of issue: April 30, 2024

Non-Confidential

Document version: 21.0

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

Document ID: SDEN-1401782

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\_1401782\_21.0\_en) was issued on April 30, 2024.

There might be a later issue at <http://developer.arm.com/documentation/SDEN-1401782>

## 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](mailto:terms@arm.com).

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Cortex-X1 (MP077), 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

<b>rOp0 implementation fixes</b>	<b>9</b>
<b>r1p0 implementation fixes</b>	<b>10</b>
<b>Introduction</b>	<b>11</b>
Scope	11
Categorization of errata	11
<b>Change Control</b>	<b>12</b>
<b>Errata summary table</b>	<b>23</b>
<b>Errata descriptions</b>	<b>31</b>
Category A	31
1468769 Vector instructions might cause deadlock under specific micro-architectural conditions	31
1609991 PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk	32
Category A (rare)	33
Category B	34
1439613 ERR0FR.INJ incorrectly indicates support for the RAS Common Fault Injection Extension	34
1467580 Branch prediction for an ERET cached in the instruction cache might cause a deadlock	35
1479207 Software Step might prevent interrupt recognition	37
1479939 Incorrect instructions might be executed	39
1492189 Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core	40
1503072 NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock	41
1515634 The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache	43
1581895 Enabling SPE might result in deadlock in some situations	44
1688305 A streaming write in the presence of a store-release instruction might result in data corruption	45
1688306 Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer	46
1688309 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	47

1821534	Atomic Store instructions to shareable write-back memory might cause memory consistency failures	48
Description		49
1827429	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB	50
1852354	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR	51
1875698	Core might generate breakpoint exception on incorrect IA	53
1941498	Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state	54
1941712	External debugger access to Debug registers might not work during Warm reset	55
1951500	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics	56
1952683	Corruption of cumulative floating point exception bits	58
2004043	Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled.	60
2004055	Incorrect programming of PMBPTR_EL1 might result in a deadlock	61
2132060	Disabling of data prefetcher with outstanding prefetch TLB miss may cause a hang	62
2242635	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion	63
2376745	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	64
2395406	Translation table walk folding into an L1 prefetch might cause data corruption	65
2712571	The core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back	66
2742426	Page crossing access that generates an MMU fault on the second page could result in a livelock	68
2772019	The core might deadlock during powerdown sequence	69
2779479	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation	70
3031174	SPE might write to pages which lack write permission at Stage-1 or Stage-2	71
3324344	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	73
Category B (rare)		74
1415185	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data	74
1543963	The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches	76
2986640	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level	77

Category C		79
1431442	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0	79
1436720	Waypoints from previous session might cause single-shot comparator match when trace enabled	80
1465945	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception Level	81
1488614	An unaligned load may initiate a prefetch request which crosses a page boundary	83
1488740	Interrupt might be taken later than architecturally mandated on exit from Debug state	84
1492301	Transient parity error in L1 instruction cache might result in missed breakpoint exception	86
1502854	TRCIDR3.CCITMIN value is incorrect	87
1511995	ESB instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE	88
1549197	PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled	89
1559545	The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception	90
1563201	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error	91
1576544	Enabling L2 cache partitioning might result in a loss of performance	92
1584334	ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort	93
1585052	A load to normal memory might trigger a prefetch request outside of the current mapped page	94
1589060	RAS error status records could log spurious corrected error	95
1643615	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect	96
1688249	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption	97
1688302	APB access to trace registers does not work during Warm reset	98
1688303	Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock	99
1688304	A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERR0MISC0_EL1 and ER0ADDR_EL1	101
1688316	ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly	102
1740838	RAS error reported could have incorrect value in ER0ADDR_EL1	103

1740840	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice	104
1740841	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB	105
1740842	The core might record incorrect INDEX into ERR0MISC0 when L0 Macro-op cache is affected by parity error	106
1740843	Instruction sampling bias exists in SPE implementation	107
1816119	Loss of CTI events during warm reset	108
1816422	The core might deadlock when an external debugger injects instructions using ITR register	109
1817659	Possible loss of CTI event	110
1817662	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	111
1827432	Watchpoint Exception on DC ZVA does not report correct address in FAR	112
1827437	Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents	113
1872190	External debug accesses in memory access mode with SCTLRL_ELx.IESB set might result in unpredictable behavior	115
1872194	Transient L2 tag double bit Errors might cause data corruption	116
1872197	ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect	117
1872200	Uncorrectable tag errors in L2 cache might cause deadlock	118
1941501	L2 data RAM may fail to report corrected ECC errors	119
1941709	IDATAN_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB	120
1941802	PFG duplicate reported faults through a Warm reset	121
1941932	The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified	122
1941935	Noncompliance with prioritization of Exception Catch debug events	123
1941938	Some corrected errors might incorrectly increment ERR0MISC0.CECC or ERR0MISC0.CECO	125
1951503	The PE might deadlock if Pseudofault Injection is enabled in Debug State	126
1983424	Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC	127
2004037	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	128
2004097	DRPS might not execute correctly in Debug state with SCTLRL_ELx.IESB set in the current EL	129

2091744	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation	130
2102456	ETM trace information records a branch to the next instruction as an N atom	131
2102758	External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register	132
2106991	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	134
2131884	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes	136
2132041	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain	137
2151897	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely	138
2242640	An SError might not be reported for an atomic store that encounters data poison	140
2280344	PMU L1D_CACHE_REFILL_OUTER is inaccurate	141
2296013	L1 Data poison is not cleared by a store	142
2341663	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk	143
2423048	Software-step not done after exit from Debug state with an illegal value in DSPSR	144
2446528	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly	145
2699191	Incorrect value reported for SPE PMU event SAMPLE_FEED	146
2699197	Reads of DISR_EL1 incorrectly return 0s while in Debug State	147
2699760	Incorrect read value for Performance Monitors Control Register	148
2708633	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state	149
2712563	Incorrect read value for Performance Monitors Configuration Register EX field	150
2764409	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	151
2817022	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	152
<b>Proprietary notice</b>		153
<b>Product and document information</b>		155
Product status		155
Product completeness status		155
Product revision status		155

## rOp0 implementation fixes

Note the following errata might be fixed in some implementations of rOp0. 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]	1468769 Vector instructions might cause deadlock under specific micro-architectural conditions
REVIDR_EL1[1]	1609991 PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk

Note that there is no change to the MIDR\_EL1 which remains at rOp0 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.



## 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]	1688305 A streaming write in the presence of a store-release instruction might result in data corruption
---------------	--

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.

# 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:

<b>Category A</b>	A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.
<b>Category A (Rare)</b>	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.
<b>Category B</b>	A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.
<b>Category B (Rare)</b>	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.
<b>Category C</b>	A minor error.

# Change Control

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The [errata summary table](#) identifies errata that have been fixed in each product revision.

## April 30, 2024: Changes in document version v21.0

ID	Status	Area	Category	Summary
<a href="#">3324344</a>	New	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing

## August 23, 2023: Changes in document version v20.0

ID	Status	Area	Category	Summary
<a href="#">3031174</a>	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
<a href="#">2986640</a>	New	Programmer	Category B (rare)	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level

## February 22, 2023: Changes in document version v19.0

ID	Status	Area	Category	Summary
<a href="#">2817022</a>	New	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM

## November 09, 2022: Changes in document version v18.0

ID	Status	Area	Category	Summary
<a href="#">2742426</a>	New	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock
<a href="#">2772019</a>	New	Programmer	Category B	The core might deadlock during powerdown sequence
<a href="#">2779479</a>	New	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation
<a href="#">2764409</a>	New	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

**August 04, 2022: Changes in document version v17.0**

ID	Status	Area	Category	Summary
<a href="#">2712571</a>	New	Programmer	Category B	Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back
<a href="#">2242640</a>	New	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison
<a href="#">2280344</a>	New	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate
<a href="#">2446528</a>	New	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly
<a href="#">2699191</a>	New	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED
<a href="#">2699197</a>	New	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State
<a href="#">2699760</a>	New	Programmer	Category C	Incorrect read value for Performance Monitors Control Register
<a href="#">2708633</a>	New	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state
<a href="#">2712563</a>	New	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field

**January 21, 2022: Changes in document version v16.0**

ID	Status	Area	Category	Summary
<a href="#">2376745</a>	New	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
<a href="#">2395406</a>	New	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption
<a href="#">2341663</a>	New	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk
<a href="#">2423048</a>	New	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR

**September 24, 2021: Changes in document version v15.0**

ID	Status	Area	Category	Summary
<a href="#">2242635</a>	New	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion
<a href="#">2296013</a>	New	Programmer	Category C	L1 Data poison is not cleared by a store

**May 13, 2021: Changes in document version v14.0**

ID	Status	Area	Category	Summary
<a href="#">1688309</a>	Updated	Programmer	Category B	Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation
<a href="#">1852354</a>	Updated	Programmer	Category B	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR
<a href="#">1875698</a>	Updated	Programmer	Category B	Core might generate breakpoint exception on incorrect IA
<a href="#">1941712</a>	Updated	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset

ID	Status	Area	Category	Summary
<a href="#">1941498</a>	Updated	Programmer	Category B	Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state
<a href="#">1951500</a>	Updated	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics
<a href="#">2004043</a>	Updated	Programmer	Category B	Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled
<a href="#">2004055</a>	Updated	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock
<a href="#">2132060</a>	New	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock
<a href="#">1740843</a>	Updated	Programmer	Category C	Instruction sampling bias exists in SPE implementation
<a href="#">1816119</a>	Updated	Programmer	Category C	Loss of CTI events during warm reset
<a href="#">1872200</a>	Updated	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock
<a href="#">1941802</a>	Updated	Programmer	Category C	PFG duplicate reported faults through a Warm reset
<a href="#">1941501</a>	Updated	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors
<a href="#">1941938</a>	Updated	Programmer	Category C	Some corrected errors might incorrectly increment ERR0MISCO.CECC or ERR0MISCO.CECO
<a href="#">1951503</a>	Updated	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State
<a href="#">1983424</a>	Updated	Programmer	Category C	Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC
<a href="#">2004037</a>	Updated	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled
<a href="#">2004097</a>	Updated	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL
<a href="#">2091744</a>	Updated	Programmer	Category C	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation
<a href="#">2102456</a>	Updated	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom
<a href="#">2102758</a>	Updated	Programmer	Category C	External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register
<a href="#">2131884</a>	New	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes
<a href="#">2132041</a>	New	Programmer	Category C	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain
<a href="#">2151897</a>	New	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely

**March 03, 2021: Changes in document version v13.0**

ID	Status	Area	Category	Summary
<a href="#">2091744</a>	New	Programmer	Category C	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation
<a href="#">2102456</a>	New	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom
<a href="#">2102758</a>	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
<a href="#">2106991</a>	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

**December 09, 2020: Changes in document version v12.0**

ID	Status	Area	Category	Summary
<a href="#">1875698</a>	Updated	Programmer	Category B	Core might generate breakpoint exception on incorrect IA
<a href="#">1951503</a>	Updated	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State

**November 13, 2020: Changes in document version v11.0**

ID	Status	Area	Category	Summary
<a href="#">2004043</a>	New	Programmer	Category B	Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled
<a href="#">2004055</a>	New	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock
<a href="#">1983424</a>	New	Programmer	Category C	Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC
<a href="#">2004037</a>	New	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled
<a href="#">2004097</a>	New	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLRL_ELx.IESB set in the current EL

## September 24, 2020: Changes in document version v10.0

ID	Status	Area	Category	Summary
<a href="#">1503072</a>	Updated	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
<a href="#">1941712</a>	New	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset
<a href="#">1941498</a>	New	Programmer	Category B	Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state
<a href="#">1951500</a>	New	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics
<a href="#">1952683</a>	New	Programmer	Category B	Corruption of cumulative floating point exception bits
<a href="#">1941802</a>	New	Programmer	Category C	PFG duplicate reported faults through a Warm reset
<a href="#">1941709</a>	New	Programmer	Category C	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB
<a href="#">1941501</a>	New	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors
<a href="#">1941932</a>	New	Programmer	Category C	The core might report incorrect fetch address to FAR_ELx when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified
<a href="#">1941935</a>	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events
<a href="#">1941938</a>	New	Programmer	Category C	Some corrected errors might incorrectly increment ERR0MISCO.CECC or ERR0MISCO.CECO
<a href="#">1951503</a>	New	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State

## June 17, 2020: Changes in document version v9.0

ID	Status	Area	Category	Summary
<a href="#">1688305</a>	Updated	Programmer	Category B	A streaming write in the presence of a store-release instruction might result in data corruption
<a href="#">1688306</a>	Updated	Programmer	Category B	Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer
<a href="#">1821534</a>	Updated	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures
<a href="#">1827429</a>	Updated	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB
<a href="#">1852354</a>	New	Programmer	Category B	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR
<a href="#">1875698</a>	New	Programmer	Category B	Core might generate breakpoint exception on incorrect IA
<a href="#">1643615</a>	Updated	Programmer	Category C	ERR0MISCO_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect
<a href="#">1688249</a>	Updated	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption

ID	Status	Area	Category	Summary
<a href="#">1688302</a>	Updated	Programmer	Category C	APB access to trace registers does not work during Warm reset
<a href="#">1688303</a>	Updated	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
<a href="#">1688304</a>	Updated	Programmer	Category C	A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERRORMISCO_EL1 and ER0ADDR_EL1
<a href="#">1688316</a>	Updated	Programmer	Category C	ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly
<a href="#">1740838</a>	Updated	Programmer	Category C	RAS error reported could have incorrect value in ER0ADDR_EL1
<a href="#">1740840</a>	Updated	Programmer	Category C	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice
<a href="#">1740841</a>	Updated	Programmer	Category C	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB
<a href="#">1740842</a>	Updated	Programmer	Category C	The core might record incorrect INDEX into ERRORMISCO when L0 Macro-op cache is affected by parity error
<a href="#">1816422</a>	Updated	Programmer	Category C	The core might deadlock when an external debugger injects instructions using ITR register
<a href="#">1817659</a>	Updated	Programmer	Category C	Possible loss of CTI event
<a href="#">1817662</a>	Updated	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
<a href="#">1827432</a>	Updated	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR
<a href="#">1827437</a>	Updated	Programmer	Category C	Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents
<a href="#">1872190</a>	New	Programmer	Category C	External debug accesses in memory access mode with SCTL*_ELx.IESB set might result in unpredictable behavior
<a href="#">1872194</a>	New	Programmer	Category C	Transient L2 tag double bit Errors might cause data corruption
<a href="#">1872197</a>	New	Programmer	Category C	ERRORMISCO_EL1.SUBARRAY, ERROSTATUS.CE and ERROSTATUS.DE values for ECC errors in the L1 data cache might be incorrect
<a href="#">1872200</a>	New	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock



## May 07, 2020: Changes in document version v8.0

ID	Status	Area	Category	Summary
<a href="#">1821534</a>	New	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures
<a href="#">1827429</a>	New	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB
<a href="#">1816119</a>	New	Programmer	Category C	Loss of CTI events during warm reset
<a href="#">1816422</a>	New	Programmer	Category C	The core might deadlock when an external debugger injects instructions using ITR register
<a href="#">1817659</a>	New	Programmer	Category C	Possible loss of CTI event
<a href="#">1817662</a>	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
<a href="#">1827432</a>	New	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR
<a href="#">1827437</a>	New	Programmer	Category C	Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents

## February 14, 2020: Changes in document version v7.0

ID	Status	Area	Category	Summary
<a href="#">1688306</a>	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
<a href="#">1688309</a>	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
<a href="#">1643615</a>	New	Programmer	Category C	ERRORMISCO_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect
<a href="#">1688249</a>	New	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption
<a href="#">1688302</a>	New	Programmer	Category C	APB access to trace registers does not work during Warm reset
<a href="#">1688303</a>	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
<a href="#">1688304</a>	New	Programmer	Category C	A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERRORMISCO_EL1 and ER0ADDR_EL1
<a href="#">1688316</a>	New	Programmer	Category C	ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly
<a href="#">1740838</a>	New	Programmer	Category C	RAS error reported could have incorrect value in ER0ADDR_EL1
<a href="#">1740840</a>	New	Programmer	Category C	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice
<a href="#">1740841</a>	New	Programmer	Category C	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB
<a href="#">1740842</a>	New	Programmer	Category C	The core might record incorrect INDEX into ERRORMISCO when L0 Macro-op cache is affected by parity error
<a href="#">1740843</a>	New	Programmer	Category C	Instruction sampling bias exists in SPE implementation

## December 11, 2019: Changes in document version v6.0

ID	Status	Area	Category	Summary
<a href="#">1688305</a>	New	Programmer	Category B	A streaming write in the presence of a store-release instruction might result in data corruption

## October 14, 2019: Changes in document version v5.0

ID	Status	Area	Category	Summary
<a href="#">1609991</a>	New	Programmer	Category A	PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk

## October 01, 2019: Changes in document version v4.0

ID	Status	Area	Category	Summary
<a href="#">1468769</a>	Updated	Programmer	Category A	Vector instructions might cause deadlock under specific micro-architectural conditions

ID	Status	Area	Category	Summary
<a href="#">1439613</a>	Updated	Programmer	Category B	ERROFR.INJ incorrectly indicates support for the RAS Common Fault Injection Extension
<a href="#">1467580</a>	Updated	Programmer	Category B	Branch prediction for an ERET cached in the instruction cache might cause a deadlock
<a href="#">1479207</a>	Updated	Programmer	Category B	Software Step might prevent interrupt recognition
<a href="#">1479939</a>	Updated	Programmer	Category B	Incorrect instructions might be executed
<a href="#">1492189</a>	Updated	Programmer	Category B	Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core
<a href="#">1503072</a>	Updated	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
<a href="#">1515634</a>	Updated	Programmer	Category B	The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache
<a href="#">1581895</a>	New	Programmer	Category B	Enabling SPE might result in deadlock in some situations
<a href="#">1415185</a>	Updated	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
<a href="#">1543963</a>	New	Programmer	Category B (rare)	The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches
<a href="#">1431442</a>	Updated	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
<a href="#">1436720</a>	Updated	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled
<a href="#">1465945</a>	Updated	Programmer	Category C	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception Level
<a href="#">1488614</a>	Updated	Programmer	Category C	An unaligned load may initiate a prefetch request which crosses a page boundary
<a href="#">1488740</a>	Updated	Programmer	Category C	Interrupt might be taken later than architecturally mandated on exit from Debug state
<a href="#">1492301</a>	Updated	Programmer	Category C	Transient parity error in L1 instruction cache might result in missed breakpoint exception
<a href="#">1502854</a>	Updated	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect
<a href="#">1511995</a>	New	Programmer	Category C	ESB instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE
<a href="#">1549197</a>	New	Programmer	Category C	PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled
<a href="#">1559545</a>	New	Programmer	Category C	The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception
<a href="#">1563201</a>	New	Programmer	Category C	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error

ID	Status	Area	Category	Summary
<a href="#">1576544</a>	New	Programmer	Category C	Enabling L2 cache partitioning might result in a loss of performance
<a href="#">1584334</a>	New	Programmer	Category C	ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort
<a href="#">1585052</a>	New	Programmer	Category C	A load to normal memory might trigger a prefetch request outside of the current mapped page
<a href="#">1589060</a>	New	Programmer	Category C	RAS error status records could log spurious corrected error

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

ID	Status	Area	Category	Summary
<a href="#">1479939</a>	New	Programmer	Category B	Incorrect instructions might be executed
<a href="#">1492189</a>	New	Programmer	Category B	Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core
<a href="#">1503072</a>	New	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
<a href="#">1515634</a>	New	Programmer	Category B	The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache
<a href="#">1488614</a>	New	Programmer	Category C	An unaligned load may initiate a prefetch request which crosses a page boundary
<a href="#">1488740</a>	New	Programmer	Category C	Interrupt might be taken later than architecturally mandated on exit from Debug state
<a href="#">1492301</a>	New	Programmer	Category C	Transient parity error in L1 instruction cache might result in missed breakpoint exception
<a href="#">1502854</a>	New	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect

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

ID	Status	Area	Category	Summary
<a href="#">1468769</a>	New	Programmer	Category A	Vector instructions might cause deadlock under specific micro-architectural conditions
<a href="#">1439613</a>	New	Programmer	Category B	ERROFR.INJ incorrectly indicates support for the RAS Common Fault Injection Extension
<a href="#">1467580</a>	New	Programmer	Category B	Branch prediction for an ERET cached in the instruction cache might cause a deadlock
<a href="#">1479207</a>	New	Programmer	Category B	Software Step might prevent interrupt recognition
<a href="#">1415185</a>	New	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
<a href="#">1431442</a>	New	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
<a href="#">1436720</a>	New	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled
<a href="#">1465945</a>	New	Programmer	Category C	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception Level

**March 27, 2019: Changes in document version v1.0**

No errata in this document version.

# Errata summary table

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

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1468769</a>	Programmer	Category A	Vector instructions might cause deadlock under specific micro-architectural conditions	r0p0	r1p0
<a href="#">1609991</a>	Programmer	Category A	PC or ELR register contents might be corrupted when an instruction fetch hits in the L0 Macro-op cache and misses in the L1 Instruction TLB generating a tablewalk	r0p0	r1p0
<a href="#">1439613</a>	Programmer	Category B	ERROFR.INJ incorrectly indicates support for the RAS Common Fault Injection Extension	r0p0	r1p0
<a href="#">1467580</a>	Programmer	Category B	Branch prediction for an ERET cached in the instruction cache might cause a deadlock	r0p0	r1p0
<a href="#">1479207</a>	Programmer	Category B	Software Step might prevent interrupt recognition	r0p0	r1p0
<a href="#">1479939</a>	Programmer	Category B	Incorrect instructions might be executed	r0p0	r1p0
<a href="#">1492189</a>	Programmer	Category B	Aarch32-only Floating Point or Advanced SIMD instruction might deadlock in processor core	r0p0	r1p0
<a href="#">1503072</a>	Programmer	Category B	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock	r0p0	r1p0
<a href="#">1515634</a>	Programmer	Category B	The core might execute multiple instructions before taking software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache	r0p0	r1p0
<a href="#">1581895</a>	Programmer	Category B	Enabling SPE might result in deadlock in some situations	r0p0	r1p0
<a href="#">1688305</a>	Programmer	Category B	A streaming write in the presence of a store-release instruction might result in data corruption	r0p0, r1p0	r1p1
<a href="#">1688306</a>	Programmer	Category B	Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer	r0p0, r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1688309</a>	Programmer	Category B	Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation	r0p0, r1p0, r1p1	r1p2
<a href="#">1821534</a>	Programmer	Category B	Atomic Store instructions to shareable write-back memory might cause memory consistency failures	r0p0, r1p0	r1p1
<a href="#">1827429</a>	Programmer	Category B	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB	r0p0, r1p0	r1p1
<a href="#">1852354</a>	Programmer	Category B	Watchpoint exception on Ld/St does not report correct address in FAR or EDWAR	r0p0, r1p0, r1p1	r1p2
<a href="#">1875698</a>	Programmer	Category B	Core might generate breakpoint exception on incorrect IA	r0p0, r1p0, r1p1	r1p2
<a href="#">1941498</a>	Programmer	Category B	Store operation that encounters multiple hits in the TLB might access regions of memory with attributes that could not be accessed at that Exception level or Security state	r0p0, r1p0, r1p1	r1p2
<a href="#">1941712</a>	Programmer	Category B	External debugger access to Debug registers might not work during Warm reset	r0p0, r1p0, r1p1	r1p2
<a href="#">1951500</a>	Programmer	Category B	Atomic instructions with acquire semantics might not be ordered with respect to older stores with release semantics	r0p0, r1p0, r1p1	r1p2
<a href="#">1952683</a>	Programmer	Category B	Corruption of cumulative floating point exception bits	r0p0	r1p0
<a href="#">2004043</a>	Programmer	Category B	Virtual to physical translation latency might not be captured for SPE records when physical address collection is disabled	r0p0, r1p0, r1p1	r1p2
<a href="#">2004055</a>	Programmer	Category B	Incorrect programming of PMBPTR_EL1 might result in a deadlock	r0p0, r1p0, r1p1	r1p2
<a href="#">2132060</a>	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2242635</a>	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2376745</a>	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	r0p0, r1p0, r1p1, r1p2	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2395406</a>	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2712571</a>	Programmer	Category B	Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2742426</a>	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2772019</a>	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2779479</a>	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">3031174</a>	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">3324344</a>	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1415185</a>	Programmer	Category B (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data	r0p0	r1p0
<a href="#">1543963</a>	Programmer	Category B (rare)	The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches	r0p0	r1p0
<a href="#">2986640</a>	Programmer	Category B (rare)	PE might incorrectly detect a Watchpoint debug event instead of a Data Abort exception on a page crossing memory access, resulting in errant entry to Debug state or routing the Data Abort exception to an incorrect Exception level	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1431442</a>	Programmer	Category C	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0	r0p0	r1p0
<a href="#">1436720</a>	Programmer	Category C	Waypoints from previous session might cause single-shot comparator match when trace enabled	r0p0	r1p0
<a href="#">1465945</a>	Programmer	Category C	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception Level	r0p0	r1p0



ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1488614</a>	Programmer	Category C	An unaligned load may initiate a prefetch request which crosses a page boundary	r0p0	r1p0
<a href="#">1488740</a>	Programmer	Category C	Interrupt might be taken later than architecturally mandated on exit from Debug state	r0p0	r1p0
<a href="#">1492301</a>	Programmer	Category C	Transient parity error in L1 instruction cache might result in missed breakpoint exception	r0p0	r1p0
<a href="#">1502854</a>	Programmer	Category C	TRCIDR3.CCITMIN value is incorrect	r0p0	r1p0
<a href="#">1511995</a>	Programmer	Category C	ESB instruction execution with a pending masked Virtual SEError might not clear HCR_EL2.VSE	r0p0	r1p0
<a href="#">1549197</a>	Programmer	Category C	PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled	r0p0	r1p0
<a href="#">1559545</a>	Programmer	Category C	The core might deadlock or detect a breakpoint at an incorrect location when a T32 instruction is affected by parity error and the next instruction is programmed as an address matching breakpoint exception	r0p0	r1p0
<a href="#">1563201</a>	Programmer	Category C	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error	r0p0	r1p0
<a href="#">1576544</a>	Programmer	Category C	Enabling L2 cache partitioning might result in a loss of performance	r0p0	r1p0
<a href="#">1584334</a>	Programmer	Category C	ESR and FAR registers could be corrupted by a speculative instruction that encounters an ECC error or external data abort	r0p0	r1p0
<a href="#">1585052</a>	Programmer	Category C	A load to normal memory might trigger a prefetch request outside of the current mapped page	r0p0	r1p0
<a href="#">1589060</a>	Programmer	Category C	RAS error status records could log spurious corrected error	r0p0	r1p0
<a href="#">1643615</a>	Programmer	Category C	ERR0MISCO_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect	r0p0, r1p0	r1p1
<a href="#">1688249</a>	Programmer	Category C	MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption	r0p0, r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1688302</a>	Programmer	Category C	APB access to trace registers does not work during Warm reset	r0p0, r1p0	r1p1
<a href="#">1688303</a>	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	r1p1
<a href="#">1688304</a>	Programmer	Category C	A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERRORMISCO_EL1 and EROADDR_EL1	r0p0, r1p0	r1p1
<a href="#">1688316</a>	Programmer	Category C	ECC error on a read of the L2 data ram entry not containing valid data might report the error incorrectly	r0p0, r1p0	r1p1
<a href="#">1740838</a>	Programmer	Category C	RAS error reported could have incorrect value in EROADDR_EL1	r0p0, r1p0	r1p1
<a href="#">1740840</a>	Programmer	Category C	Some load instructions executed in Debug state through the Instruction Transfer Register might execute twice	r0p0, r1p0	r1p1
<a href="#">1740841</a>	Programmer	Category C	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB	r0p0, r1p0	r1p1
<a href="#">1740842</a>	Programmer	Category C	The core might record incorrect INDEX into ERRORMISCO when L0 Macro-op cache is affected by parity error	r0p0, r1p0	r1p1
<a href="#">1740843</a>	Programmer	Category C	Instruction sampling bias exists in SPE implementation	r0p0, r1p0, r1p1	r1p2
<a href="#">1816119</a>	Programmer	Category C	Loss of CTI events during warm reset	r0p0, r1p0, r1p1	r1p2
<a href="#">1816422</a>	Programmer	Category C	The core might deadlock when an external debugger injects instructions using ITR register	r0p0, r1p0	r1p1
<a href="#">1817659</a>	Programmer	Category C	Possible loss of CTI event	r0p0, r1p0	r1p1
<a href="#">1817662</a>	Programmer	Category C	A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data	r0p0, r1p0	r1p1
<a href="#">1827432</a>	Programmer	Category C	Watchpoint Exception on DC ZVA does not report correct address in FAR	r0p0, r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1827437</a>	Programmer	Category C	Memory uploads and downloads via memory access mode within Debug state can fail to accurately read or write memory contents	r0p0, r1p0	r1p1
<a href="#">1872190</a>	Programmer	Category C	External debug accesses in memory access mode with SCTLR_ElX.IESB set might result in unpredictable behavior	r0p0, r1p0	r1p1
<a href="#">1872194</a>	Programmer	Category C	Transient L2 tag double bit Errors might cause data corruption	r0p0, r1p0	r1p1
<a href="#">1872197</a>	Programmer	Category C	ERRORMISCO_EL1.SUBARRAY, ERROSTATUS.CE and ERROSTATUS.DE values for ECC errors in the L1 data cache might be incorrect	r0p0, r1p0	r1p1
<a href="#">1872200</a>	Programmer	Category C	Uncorrectable tag errors in L2 cache might cause deadlock	r0p0, r1p0	r1p1
<a href="#">1941501</a>	Programmer	Category C	L2 data RAM may fail to report corrected ECC errors	r0p0, r1p0, r1p1	r1p2
<a href="#">1941709</a>	Programmer	Category C	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1941802</a>	Programmer	Category C	PFG duplicate reported faults through a Warm reset	r0p0, r1p0, r1p1	r1p2
<a href="#">1941932</a>	Programmer	Category C	The core might report incorrect fetch address to FAR_ElX when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1941935</a>	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1941938</a>	Programmer	Category C	Some corrected errors might incorrectly increment ERRORMISCO.CECR or ERRORMISCO.CECO	r0p0, r1p0, r1p1	r1p2
<a href="#">1951503</a>	Programmer	Category C	The PE might deadlock if Pseudofault Injection is enabled in Debug State	r0p0, r1p0, r1p1	r1p2
<a href="#">1983424</a>	Programmer	Category C	Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR_EL1.FSC	r0p0, r1p0, r1p1	r1p2
<a href="#">2004037</a>	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	r0p0, r1p0, r1p1	r1p2

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2004097</a>	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL	r0p0, r1p0, r1p1	r1p2
<a href="#">2091744</a>	Programmer	Category C	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation	r0p0, r1p0, r1p1	r1p2
<a href="#">2102456</a>	Programmer	Category C	ETM trace information records a branch to the next instruction as an N atom	r0p0, r1p0, r1p1	r1p2
<a href="#">2102758</a>	Programmer	Category C	External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register	r0p0, r1p0, r1p1	r1p2
<a href="#">2106991</a>	Programmer	Category C	An execution of MSR instruction might not update the destination register correctly when an external debugger initiates an APB write operation to update debug registers	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2131884</a>	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes	r0p0, r1p0, r1p1	r1p2
<a href="#">2132041</a>	Programmer	Category C	OSECRR_EL1/EDECCR is incorrectly included in the Warm Reset domain	r0p0, r1p0, r1p1	r1p2
<a href="#">2151897</a>	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2242640</a>	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2280344</a>	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2296013</a>	Programmer	Category C	L1 Data poison is not cleared by a store	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2341663</a>	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2423048</a>	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2446528</a>	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly	r0p0, r1p0, r1p1, r1p2	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2699191</a>	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2699197</a>	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2699760</a>	Programmer	Category C	Incorrect read value for Performance Monitors Control Register	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2708633</a>	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2712563</a>	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2764409</a>	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2817022</a>	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	r0p0, r1p0, r1p1, r1p2	Open

# Errata descriptions

## Category A

1468769

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

### Status

Fault Type: Programmer Category A

Fault Status: Present in r0p0. Fixed in r1p0.

### Description

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

### Configurations Affected

This erratum affects all configurations.

### Conditions

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

### Implications

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

### Workaround

There is no workaround.

## 1609991

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

#### Status

Fault Type: Programmer Category A

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

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

#### Workaround

There is no workaround.

## Category A (rare)

There are no errata in this category.



## Category B

1439613

### ERROFR.INJ incorrectly indicates support for the RAS Common Fault Injection Extension

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

The ERROFR register reports INJ=0x1. This indicates support for the RAS Common Fault Injection Extension, which is not supported by the core.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs when a read from ERROFR is executed.

#### Implications

The core incorrectly reports support for the RAS Common Fault Injection Model Extension.

#### Workaround

This erratum can be avoided by ignoring the value in the ERROFR.INJ register field and treating it as 0x0.

## 1467580

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

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

MSR S3\_6\_c15\_c8\_1,x0 ; MSR CPUPCR\_EL3, X0  
ISB

## 1479207

### Software Step might prevent interrupt recognition

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### 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 MDSCR\_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.

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).

#### Implications

#### Case A:

Arm believes that for this product, MDSCR\_EL1.KDE is not set to 1 by deployed devices in the field and is only used when debugging the system software during initial product development. In these cases, the effect of the erratum is for interrupts to be disabled even after switching to other software contexts that are not being debugged as part of the system software debugging. Arm believes that a workaround does not need to be deployed for the situation where MDSCR\_EL1.KDE==1, and a workaround is not available.

Some devices are expected to run an operating system at EL2 with HCR\_EL2.E2H set to 1. The implication of this erratum for such a system is that single-stepping of an untrusted user application at ELO can lead to subsequent execution not recognizing interrupts where it should, leading to errant behavior. The software workaround described below can be deployed in the operating system at EL2 to prevent single-stepping of untrusted user applications from triggering this erratum.

#### Case B:

Unconventional code involving the Software Stepping of the disabling instruction is not expected to be encountered, therefore no workaround is required.

## Workaround

When Software Step is used to debug an application under an operating system running at EL2 with HCR\_EL2.E2H set to 1, the software workaround involves explicitly triggering a Software Step exception with modifications to the system call exception handler code and Software Step exception handler code. This entails setting MDSCR\_EL1.KDE and MDSCR\_EL1.SS and clearing PSTATE.D to trigger a Software Step exception from the system call handler. The Software Step handler then sets SPSR\_ELx.D before returning back to the system call handler, where MDSCR\_EL1.KDE and MDSCR\_EL1.SS are restored to their original values.

If a workaround is required when MDSCR\_EL1.KDE is set to 1, then please contact Arm.

## 1479939

### Incorrect instructions might be executed

#### Status

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

#### Description

Incorrect instructions might be executed in AArch64 state.

#### Configurations Affected

This erratum affects all configurations with CORE\_POP\_RAM set to TRUE.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum can be avoided by setting CPUACTLR\_EL1[13] to 1 to disable a performance feature. This should be done before enabling the MMU.

**1492189**

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

### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

### Description

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

### Configurations Affected

This erratum affects all configurations.

### Conditions

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

### Implications

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

### Workaround

The workaround is to set CPUACTLR5\_EL1[8] to 1'b1. The workaround might result in a small increase in core power consumption.

## 1503072

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

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

;; Inserts a DMB SY before STREX imm offset
LDR x0,=0x1
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e840000
MSR S3_6_c15_c8_2,x0
```



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

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

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

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

;; Synchronize to enable patches
ISB
```

## 1515634

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

### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

### Description

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

### Configurations Affected

This erratum affects all configurations.

### Conditions

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

### Implications

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

### Workaround

Set CPUACTLR\_EL1[11] to one, which flushes the L0 Macro-op cache for all context synchronization events.

## 1581895

### Enabling SPE might result in deadlock in some situations

#### Status

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

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum can be avoided by disabling SPE, by setting PMBLIMITR\_EL1.E = 0. However, the deadlock is found to occur rarely, therefore SPE could be enabled and used for prototyping purposes.

## 1688305

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum can be avoided by setting CPUACTLR2\_EL1[1] to 1, which prevents the store-release from being dispatched before it is the oldest.

## 1688306

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

## 1688309

**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, r1p1. Fixed in r1p2.

### 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.

**1821534**

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

### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

## Description

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

## Configurations Affected

This erratum affects all configurations.

## Conditions

1. PEO executes Atomic Store instruction that hits in the L1 data cache and L2 cache in the Shared state.
2. PEO 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. PEO invalidating snoop to the L1 data cache is delayed due to internal queueing.

## Implications

If the above conditions are met, PEO 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.



## 1827429

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations affected

All configurations are affected.

#### Conditions

All of the following conditions must be met:

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

#### Implications

If the above conditions are met, then the MMU might return stale translation for a subsequent access. The transient single-bit ECC error will be reported in `ERRORMISCO_EL1` register.

#### Workaround

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

## 1852354

### 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, r1p1. Fixed in r1p2.

#### 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:

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.

## 1875698

### Core might generate breakpoint exception on incorrect IA

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, and r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This exception might occur when:

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

#### Implications

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

#### Workaround

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

Note: this erratum was previously published with a different workaround, which entailed setting CPUACTLR\_EL1[21] to 1'b1. That workaround should only be applied to r0p0 hardware.

## 1941498

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum can be avoided by setting CPUECTLR\_EL1[8] to 1. There is a small performance cost (<0.5%) for setting this bit.

## 1941712

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

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

#### Implications

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

#### Workaround

There is no workaround.

## 1951500

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

#### Status

Fault Type: Programmer Category B.

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

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

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

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

ISB

Note that there is no workaround provided for rOp0 hardware. Please contact Arm support for further details.



## 1952683

### Corruption of cumulative floating point exception bits

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions:

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

#### Implications

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

#### Workaround

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

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

```
LDR x0,=0xEE10A10
MSR S3_6_c15_c8_2,x0
LDR x0,=0xFFEF0FFF
MSR S3_6_c15_c8_3,x0
LDR x0,=0x0010F000
MSR S3_6_c15_c8_4,x0
LDR x0,=0x0010F000
MSR S3_6_c15_c8_5,x0
LDR x0,=0x40000080023ff
MSR S3_6_c15_c8_1,x0

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

ISB
```

## 2004043

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Physical address collection is disabled for SPE records at the appropriate EL by setting PMSCR\_EL1.PA=0 or PMSCR\_EL2.PA=0.

#### Implications

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

#### Workaround

Where it is acceptable to capture the physical address, this erratum can be avoided by enabling physical address sampling, by setting PMSCR\_EL1.PA = 1 and PMSCR\_EL2.PA = 1.

## 2004055

### Incorrect programming of PMBPTR\_EL1 might result in a deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

When PMBPTR\_EL1 is incorrectly programmed to be equal to or greater than PMBLIMITR\_EL1, then under certain conditions, the CPU might deadlock.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

The erratum occurs under the following conditions:

1. SPE is enabled.
2. PMBSR\_EL1.S = 0, indicating PMBIRQ is not asserted.
3. PMBPTR\_EL1 is programmed to be equal to or greater than PMBLIMITR\_EL1.

#### Implications

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

#### Workaround

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

A hypervisor at EL2 can configure MDCR\_EL2.E2PB to trap EL1 accesses to PMBPTR\_EL1, PMBLIMITR\_EL1, and PMBSR\_EL1. The hypervisor can mediate these accesses and maintain a shadow copy of PMBLIMITR\_EL1 such that the physical PMBLIMITR\_EL1 register has PMBLIMITR\_EL1.E clear whenever PMBPTR\_EL1.PTR >= PMBLIMITR\_EL1.LIMIT.

Firmware at EL3 can configure MDCR\_EL3.NSPB to disable SPE in the active security state and trap erroneous EL1/EL2 accesses to the SPE registers. Software written correctly should not access the SPE registers in this case.

## 2132060

### Disabling of data prefetcher with outstanding prefetch TLB miss may cause a hang

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

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

#### Implications

If the above conditions are met, a hang may occur on the next context switch.

#### Workaround

- Workaround option 1:  
If the following code surrounds the MSR, it will prevent the erratum from happening:
  - tlbi (to blind address) local version (does not have to be broadcast)
  - dsb
  - isb
  - MSR CPUECTLR - disabling the prefetcher
  - isb
- Workaround option 2:  
Place the data prefetcher in the most conservative mode instead of disabling it. This will greatly reduce prefetches but not eliminate them. This is accomplished by writing the following bits to the value indicated:
  - ecltr[7:6], PF\_MODE = 2'b11

## 2242635

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

```
LDR x0,=0x5
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0x10F600E000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0x10FF80E000
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x8000000003FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
```

ISB

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

## 2376745

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

Set CPUACTLR2\_EL1[0] to 1 to force PLDW/PRFM 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.

## 2395406

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open

#### Description

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

#### Configurations Affected

This erratum affects all configurations

#### Conditions

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

#### Implications

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

#### Workaround

Disable folding of demand requests into older prefetches with L2 miss requests outstanding by setting CPUACTLR2\_EL1[40] to 1.



## 2712571

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

If a core is fetching instructions from memory while stage 1 translation is disabled and instruction cache is disabled, the core ignores Stage 2 forced Write-Back indication programmed by HCR\_EL2.FWB and make Non-cacheable, Normal memory request. This may cause the core to fetch stale data from memory subsystem.

#### Configurations Affected

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

#### Conditions

The erratum occurs if all the following conditions apply:

- The *Processing Element* (PE) is using EL1 translation regime.
- Stage 2 translation is enabled (HCR\_EL2.VM=1).
- Stage 1 translation is disabled (SCTLR\_EL1.M=0).
- Instruction cache is enabled from EL2 (HCR\_EL2.ID=0).
- Instruction cache is disabled from EL1 (SCTLR\_EL1.I=0).

#### Implications

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

#### Workaround

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

## 2742426

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

This erratum occurs under all of the following conditions:

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

#### Implications

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

#### Workaround

This erratum can be avoided by setting CPUACTLR5\_EL1[56:55] to 2'b01.

## 2772019

### The core might deadlock during powerdown sequence

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. Error detection and correction is enabled through ERXCTLR\_EL1.ED=1.
2. PE executes more than 24 writes to Device-nGnRnE or Device-nGnRE memory.
3. PE executes powerdown sequence as described in the *Technical Reference Manual* (TRM).

#### Implications

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

#### Workaround

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

## 2779479

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

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

#### Implications

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

#### Workaround

The erratum can be avoided by setting CPUACTLR3\_EL1[47]. Setting this chicken bit might have a small impact on power and negligible impact on performance.

## 3031174

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1 and r1p2. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations that support SPE.

#### Conditions

This erratum occurs under the following conditions:

1. The SPE buffer is enabled.
2. Registers PMBPTR\_EL1 and PMBLIMITR\_EL1 are configured to include a virtual address VA\_X.
3. A valid Stage-1 translation exists for the virtual address VA\_X.
4. If Stage-2 is enabled, a valid Stage-2 translation exists for the intermediate physical address IPA\_X for the virtual address VA\_X.
5. At least one of the following conditions is true:
  - a. The Stage-1 translation for VA\_X lacks write permission.
  - b. The Stage-2 translation for IPA\_X lacks write permission.
6. None of the following apply:
  - a. Stage-1 hardware dirty bit management is enabled.
  - b. Stage-2 is enabled, and Stage-2 hardware dirty bit management is enabled.

#### Implications

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

#### Workaround

No hardware workaround is available.

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

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

No current software is expected to have this problem.

## 3324344

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

The erratum occurs if the following condition applies:

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

#### Implications

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

#### Workaround

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

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



## Category B (rare)

1415185

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

#### Status

Fault Type: Programmer Category B (Rare)

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

1. The core is executing at AArch32 at EL0.
2. An MRRC to CNTPCT, CNTVCT, CNTP\_CVAL, or CNTV\_CVAL is executed.

#### Implications

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

#### Workarounds

The erratum can be avoided by trapping MRC/MCR/MRRC/MCRR accesses in AArch32 to the affected registers and doing the equivalent code sequence in the trap handler.

To trap the CNT\* accesses, set CNTKCTL\_EL1.{ELOPTEN, ELOVTEN, ELOVCTEN, ELOPCTEN} to 0. If HCR\_EL2.{E2H,TGE}={1,1} then set CNTHCTL\_EL2.{ELOPTEN, ELOVTEN, ELOVCTEN, ELOPCTEN} to 0. The following registers will be trapped:

- CNTP\_CTL.
- CNTP\_CVAL.
- CNTP\_TVAL.

- CNTV\_CTL.
- CNTV\_CVAL.
- CNTV\_TVAL.
- CNTPCT.
- CNTVCT.
- CNTFRQ.

## 1543963

### The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches

#### Status

Fault Type: Programmer Category B (Rare)

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

When the core executes a direct branch that has been recently modified, associated with prefetch speculation protection, the core might fetch a stale instructions from the L0 Macro-op cache which violates the ordering of instruction fetches.

#### Configurations Affected

This erratum affects all multi-core configurations.

#### Conditions

1. The core is in AArch64 mode.
2. The modifying core changes instructions at address A.
3. The modifying core executes cache maintenance and synchronization instructions to make address A visible to all cores in the inner shareable domain.
4. A direct branch or a NOP is substituted with a direct branch targeting address A on the modifying core.
5. The executing core fetches the branch and correctly predicts the destination of the direct branch based on stale history due to ASID or VMID reuse.
6. Stale instructions are fetched from the L0 Macro-op cache, on the executing core, instead of the modified instructions at address A.

#### Implications

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

#### Workaround

This erratum can be avoided by invalidating branch history before reusing any ASID for a new address space. This can be done by ensuring 124 ASIDs are selected before any ASID is reused.

## 2986640

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

### Status

Fault Type: Programmer Category B (Rare)

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

### Description

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

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

### Configurations affected

This erratum affects all configurations.

### Conditions

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

### Implications

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

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

- If MDCR\_EL2.TDE == 0, a stage 2 Data Abort might result in a Data Abort exception taken erroneously to EL1.
  - The rarity of PE internal timings required to exhibit this bug is comparable to *Reliability, Availability, and Serviceability* (RAS) error FIT rates. Expected outcome is a kernel panic that will kill the process.

- If MDCR\_EL2.TDE == 1, a stage 1 Data Abort might result in a Data Abort exception taken erroneously to EL2.
  - This scenario is containable within a hypervisor via the software workaround outlined below.

## Workaround

There is no complete workaround for this erratum. A partial software workaround addresses the more serious scenario of a stage 1 Data Abort resulting in a Data Abort exception taken erroneously to EL2 without updating HPFAR\_EL2.

EL2 can protect against this case as follows:

- Reserve one bit of IPA space so that VTCR\_EL2.PS is never the maximum supported.
- Write all 1's to HPFAR\_EL2[63:0] before entering EL1 or EL0.
- Exceptions to EL2 due to this erratum that should have set HPFAR\_EL2 will instead use an out of range IPA. The guest should be restarted as the conditions for this erratum are rare and are not likely to be encountered again.

## Category C

1431442

### TLBI does not treat upper ASID bits as zero when TCR\_EL1.AS is 0

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

TLBI instructions are not treating ASID[15:8] as zero when TCR\_EL1.AS=0, as specified in the Arm Architecture Reference Manual. In this configuration, the bits are RES0, which should be written to zero by software, and ignored by hardware.

#### Configurations Affected

The erratum affects all configurations.

#### Conditions

1. TCR\_EL1.AS=0.
2. A TLBI is executed with ASID[15:8] not equal to zero.

#### Implications

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

#### Workaround

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

## 1436720

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

## 1465945

### 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. Fixed in r1p0.

#### Description

If the interconnect does not support atomic memory operations, then instructions which try to perform these operations to Non-cacheable or Device memory take an IMPLEMENTATION DEFINED fault with Data Fault Status Code of ESR\_ELx.DFSC = 0b110101.

Under the following conditions, this fault has to be routed to EL1 because Stage1 fault takes priority over Stage2 fault:

- The PE is executing at Non-secure EL0 or EL1.
- Stage2 translation is enabled.
- SCTLR\_EL1.C bit forces Stage1 translation to Normal memory to be Non-cacheable.
- HCR\_EL2.CD bit forces Stage2 translation to Normal memory to be Non-cacheable.

Because of this erratum, the fault is incorrectly routed to EL2.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The interconnect does not support atomic operations.
2. The PE is executing at Non-secure EL0 or EL1.
3. SCTLR\_EL1.C is 0, forcing Stage1 translation to Normal memory to be Non-cacheable.
4. HCR\_EL2.CD is 1, forcing Stage2 translation to Normal memory to be Non-cacheable.
5. There is an atomic instruction to Normal memory.

#### Implications

If the above conditions are met, then the IMPLEMENTATION DEFINED fault with Data Fault Status Code of ESR\_ELx.DFSC = 0b110101 is routed to EL2.

#### Workaround



There is no workaround for this erratum.

## 1488614

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

The erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

## 1488740

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions:

Case A:

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

Case B:

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

change (to AArch32) and with an interrupt unmasked (PSTATE.A==0 | PSTATE.I==0 | PSTATE.F==0).

5. Exit Debug state with an interrupt pending.

## Implications

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

## Workaround

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

## 1492301

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations 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. A transient parity error occurs when reading the L1 instruction cache near the breakpoint location.
4. At least one `RAMINDEX` operation targeting the L1 instruction cache in the core with the breakpoint is outstanding.

#### Implications

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

#### Workaround

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

## 1502854

### TRCIDR3.CCITMIN value is incorrect

#### Status

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

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

## 1511995

### ESB instruction execution with a pending masked Virtual SError might not clear HCR\_EL2.VSE

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

If a Virtual SError is pending and masked at the current Exception level when an Error Synchronization Barrier (ESB) instruction is executed, then the VDISR\_EL2 update occurs properly, but sometimes the clearing of HCR\_EL2.VSE might not occur. This failure to clear HCR\_EL2.VSE can only occur when the Virtual SError is masked.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions:

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

#### Implications

If the above conditions are met, then under specific microarchitectural timing conditions HCR\_EL2.VSE might not be cleared to 0, which the Arm architecture requires. This might result in spurious Virtual SErrors. Under all circumstances, the Virtual SError syndrome from VSESR\_EL2 is correctly recorded in VDISR\_EL2, and VDISR\_EL2.A is correctly set to 1.

#### Workaround

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

**1549197****PDP Issue Queue Virtual Size Reduction remains Engaged when PDP is Disabled****Status**

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

**Description**

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

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

1. The core enables Performance defined power (PDP) optimization.
2. PDP engages power savings, due to low utilization of parts of issue queues.
3. While engaged PDP is disabled, either by pin control or by system register access to CPUPPMCR\_EL3

**Implications**

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

**Workaround**

There are 2 options to workaround this issue.

1. Never enable PDP feature by ensuring that boot code never sets CPUPPMCR\_EL3[49:48] to a non-zero value.
2. Once PDP feature is enabled, never disable it. If use of PDP feature is desired, boot code must set CPUPPMCR\_EL3[49:48] to 2'b10 and never change.



## 1559545

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

### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

### Description

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

### Configurations Affected

This erratum affects the configuration with `CORE_CACHE_PROTECTION = 1`.

### Conditions

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

### Implications

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

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

### Workaround

This erratum has no workaround.

## 1563201

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

### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

### Description

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

### Configurations Affected

This erratum affects the configuration with `CORE_CACHE_PROTECTION = 1`.

### Conditions

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

### Implications

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

### Workaround

This erratum has no workaround.

## 1576544

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

#### Status

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

#### Description

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

#### Configurations Affected

The erratum affects all configurations.

#### Conditions

L2 cache partitioning is enabled by setting either CPUECTLR\_EL1[60:58] or CPUECTLR\_EL1[57:55] to a non-zero value.

#### Implications

Setting CPUECTLR\_EL1[60:58] to a non-zero-value restricts the number of ways that can be allocated by lines containing instruction. Similarly, setting CPUECTLR\_EL1[57:55] to a non-zero value restricts the number of ways that can be allocated by lines containing data. This might have an impact on performance.

#### Workaround

This erratum can be avoided by disabling L2 cache partitioning by setting either CPUECTLR\_EL1[60:58] or CPUECTLR\_EL1[57:55] to zero.

## 1584334

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

The erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

There is no workaround for this erratum.

## 1585052

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

## 1589060

### RAS error status records could log spurious corrected error

#### Status

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

#### Description

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

#### Configurations Affected

The erratum affects configurations with `CORE_CACHE_PROTECTION` set to `TRUE`.

#### Conditions

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

#### Implications

If the above conditions are met, then:

1. `ERRMISCO_EL1.CECO` might be incremented for both the real SBE and a spurious error.
2. `ERRMISCO_EL1.OFO` might be set to indicate the corrected error count, other, has overflowed.
3. `ERRMISCO_EL1.CECR` might be incremented for both the real SBE and a spurious error.
4. `ERRMISCO_EL1.OFR` might be set to indicate the corrected error count, repeat, has overflowed.
5. `nFAULTIRQ[0]` might be set by the spurious corrected error if either counter has overflowed and `ERROCTLR_EL1.CFI` is set.

#### Workaround

There is no workaround for this erratum. The effects can be mitigated by allowing for a larger number of corrected errors to cause overflow conditions in the `ERRMISCO_EL1.CECO` and `ERRMISCO_EL1.CECR` fields.

## 1643615

### ERR0MISCO\_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. Fixed in r1p1.

#### Description

Under certain conditions, the ERR0MISCO\_EL1.SUBARRAY value recorded for ECC errors in the L1 data cache might be incorrect.

#### Configurations Affected

This erratum affects configurations with CORE\_CACHE\_PROTECTION set to TRUE.

#### 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 ERR0MISCO\_EL1.SUBARRAY might have an incorrect value. The remaining fields of the ERR0MISCO\_EL1 register remain correct.

#### Workaround

There is no workaround for this erratum.

## 1688249

### 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. Fixed in r1p1.

#### 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 ELO.
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 ELO, 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.



## 1688302

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

## 1688303

### 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. Fixed in r1p1.

#### 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 configurations with CORE\_CACHE\_PROTECTION set to TRUE.

#### Conditions

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

#### Implications

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

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

#### Workaround

Software should avoid the use of cache maintenance operations by set/way. A hypervisor should trap these instructions by setting HCR\_EL2.TSW = 1 and emulate the instructions with equivalent cache maintenance operations by virtual address for the entire address space of the guest.

## 1688304

### A load observing a double-bit ECC error after a snoop detected a single-bit ECC error might report incorrect values in ERRORMISCO\_EL1 and EROADDR\_EL1

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

Under certain conditions, a load that observes a double-bit ECC error (DBE) in the L1 data cache data RAM after a snoop observed a single-bit ECC error (SBE) might result in incorrect information being recorded in the ERRORMISCO\_EL1 and EROADDR\_EL1 registers.

#### Configurations Affected

This erratum affects configurations with CORE\_CACHE\_PROTECTION set to TRUE.

#### Conditions

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

#### Implications

If the above conditions are met, then ERRORMISCO\_EL1.SUBARRAY, ERRORMISCO\_EL1.WAY, and EROADDR\_EL1 might have incorrect values. The remaining fields of the ERRORMISCO\_EL1 register remain correct.

#### Workaround

There is no workaround for this erratum.

## 1688316

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

All configurations with CORE\_CACHE\_PROTECTION enabled.

#### Conditions

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

#### Implications

If the conditions occur the error is typically not reported.

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

#### Workaround

No workaround is required.

## 1740838

### RAS error reported could have incorrect value in ERR0ADDR\_EL1

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

Under certain conditions, capacity eviction of a line which single or double bit ECC error is in the process of being reported could end up corrupting the value in ERR0ADDR\_EL1 register.

#### Configurations Affected

The erratum affects configurations with CORE\_CACHE\_PROTECTION set to TRUE.

#### Conditions

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

#### Implications

If the above conditions are met, ERR0ADDR\_EL1 could have incorrect value. In the case of a single bit error, the data will be corrected and in the case of a double bit error data is written out as poisoned.

#### Workaround

There is no workaround for this erratum.

## 1740840

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

A workaround is only needed if there is any possibility of connecting an external debugger to the core. If that possibility exists, setting CPUACTLR3\_EL1[47] in the boot sequence will prevent this behavior. There is no performance impact associated with setting this bit, but there is a potential (workload dependent) power increase of approximately 1.5% total core power.

## 1740841

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum has no workaround.



## 1740842

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations Affected

This erratum affects the configuration with CORE\_CACHE\_PROTECTION = 1.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum has no workaround.

## 1740843

### Instruction sampling bias exists in SPE implementation

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, and r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. SPE configured and utilized on PE.

#### Implications

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

#### Workaround

No hardware workaround.

## 1816119

### Loss of CTI events during warm reset

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, and r1p1. Fixed in r1p2.

#### 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.

## 1816422

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

The core might deadlock if the above conditions are satisfied.

#### Workaround

This erratum has no workaround.

## 1817659

### Possible loss of CTI event

#### Status

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

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

CTI events might be dropped.

#### Workaround

This erratum has no workaround.

## 1817662

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

### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

### Description:

A memory mapped write to PMSSRR at offset 0x6f4 might configure the Cycle counter and/or Performance Monitor event counters to be reset along with reset of corresponding overflow status bits in the PMOVSRR 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\_ELO if x=31 or PMEVCNTR<x>\_ELO if x = [0,5]) will reset after a snapshot is taken. Further, the corresponding bit in the PMOVSRR\_ELO 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.

## 1827432

### Watchpoint Exception on DC ZVA does not report correct address in FAR

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

If the watchpoint address targets a lower portion of a cache line, but not all of the cache line, and the address target of the DC ZVA falls in the upper portion of the cache line that the watchpoint does not target the FAR 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.

#### 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.

## 1827437

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

For memory upload:

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

For memory download:

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



## Implications

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

## Workaround

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

1. Perform the memory upload or download operations with the debugger's FAST\_MEMORY\_ACCESS disabled. This can impact the performance of memory upload and download operations in Debug state, resulting in slight visible delays in the debugger user interface on memory upload and longer download times.  
or
2. Set CPUACTLR3\_EL1[47] in the boot sequence to prevent the faulty behavior. There is no performance impact associated with setting this bit, but there is a potential (workload dependent) power increase of approximately 1.5% total core power.

## 1872190

### External debug accesses in memory access mode with SCTL<sub>R</sub>\_EL<sub>x</sub>.IESB set might result in unpredictable behavior

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

In Debug state with SCTL<sub>R</sub>\_EL<sub>x</sub>.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. SCTL<sub>R</sub>\_EL<sub>x</sub>.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 SCTL<sub>R</sub>\_EL<sub>x</sub>.IESB before performing memory uploads or downloads in Debug state using memory access mode.

## 1872194

### Transient L2 tag double bit Errors might cause data corruption

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations affected

All configurations with CORE\_CACHE\_PROTECTION enabled.

#### Conditions

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

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

#### Implications

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

#### Workaround

There is no workaround.

## 1872197

### ERR0MISCO\_EL1.SUBARRAY, ERROSTATUS.CE and ERROSTATUS.DE values for ECC errors in the L1 data cache might be incorrect

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

Under certain conditions, the ERR0MISCO\_EL1.SUBARRAY, ERROSTATUS.CE and ERROSTATUS.DE values recorded for ECC errors in the L1 data cache might be incorrect.

#### Configurations affected

This erratum affects configurations with CORE\_CACHE\_PROTECTION set to TRUE.

#### 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 ERR0MISCO\_EL1.SUBARRAY, ERROSTATUS.CE and ERROSTATUS.DE might have an incorrect values.

#### Workaround

There is no workaround for this erratum.

## 1872200

### Uncorrectable tag errors in L2 cache might cause deadlock

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0. Fixed in r1p1.

#### Description

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

#### Configurations affected

This erratum affects configurations with `CORE_CACHE_PROTECTION TRUE`.

#### Conditions

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

#### Implications

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

#### Workaround

There is no workaround for this erratum.

## 1941501

### L2 data RAM may fail to report corrected ECC errors

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

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

#### Implications

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

#### Workaround

There is no workaround.

## 1941709

### IDATAn\_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

After implementation-defined RAMINDEX register is programmed to initiate direct memory access to internal memory for Instruction TLB, implementation-defined IDATAn\_EL3 value represents unpredictable value.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

If the above conditions are met, IDATAn\_EL3 register might represent incorrect value for Translation regime, VMID, ASID, and VA[48:21].

#### Workaround

There is no workaround.

## 1941802

### PFG duplicate reported faults through a Warm reset

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

1. ERROPFGCDN is set with a non-zero countdown value.
2. ERROPFGCTL is set to generate a pseudo-fault with ERROPFGCTL.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 ERROPFGCTL control bits before asserting a Warm reset.



## 1941932

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

### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

### Description

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

### Configurations Affected

All configurations are affected.

### Conditions

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

### Implications

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

### Workaround

There is no workaround.

## 1941935

### Noncompliance with prioritization of Exception Catch debug events

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

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

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

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

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

## 1941938

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

No workaround is expected to be required.

## 1951503

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, and r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

OR

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

#### Implications

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

#### Workaround

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

## 1983424

### Incorrect fault status code might be reported in Statistical Profiling Extension register PMBSR\_EL1.FSC

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

A statistical profiling buffer translation request which encounters multiple hits in the TLB might report an incorrect fault status code in PMBSR\_EL1.FSC.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

If the above conditions are met, then the fault status code reported in PMBSR\_EL1.FSC might incorrectly indicate an illegal or incorrect fault status code instead of the correct TLB Conflict fault code.

#### Workaround

There is no workaround.

## 2004037

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Timestamp capture is enabled for SPE records at the appropriate EL by setting PMSCR\_EL1.TS or PMSCR\_EL2.TS.

#### Implications

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

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

#### Workaround

There is no workaround.

## 2004097

### DRPS might not execute correctly in Debug state with SCTL\_R\_ELx.IESB set in the current EL

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

In Debug state with SCTL\_R\_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. SCTL\_R\_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 SCTL\_R\_ELx.IESB followed by the insertion of an **ISB** and an **ESB** instruction in code before the **DRPS** instruction.



## 2091744

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

The erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

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

#### Implications

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

#### Workaround

This erratum has no workaround.

## 2102456

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This issue might occur when:

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

#### Implications

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

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

#### Workaround

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

## 2102758

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

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

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

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

#### Workaround

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

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

## 2106991

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

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

cycle.

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

## Implications

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

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

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

## Workaround

No workaround is required for this erratum.

## 2131884

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, and r1p1. Fixed in r1p2.

#### Description

Collision information captured by PMBSR\_EL1.COLL might be lost under certain circumstances, when the buffer management interrupt is raised.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

If the above conditions are met, the collision indicator in PMBSR\_EL1 is incorrectly set to 0, following the 2nd SEI error. PMBSR\_EL1 does capture and set the "Data Loss" (DL) indicator and all the other PMBSR\_EL1 fields correctly.

#### Workaround

There is no workaround for this erratum.

## 2132041

### OSECCR\_EL1/EDECCR is incorrectly included in the Warm Reset domain

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

OSECCR\_EL1/EDECCR is incorrectly included in the Warm Reset domain. If a Warm Reset occurs, then the value in this register will be lost.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Warm Reset is asserted.

#### Implications

If the above conditions are met, then the value in OSECCR\_EL1/EDECCR will be lost.

#### Workaround

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



## 2151897

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

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

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

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

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

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

#### Workaround

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

## 2242640

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

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

#### Implications

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

#### Workaround

This erratum has no workaround.

## 2280344

### PMU L1D\_CACHE\_REFILL\_OUTER is inaccurate

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. 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.

## 2296013

### L1 Data poison is not cleared by a store

#### Status

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

## 2341663

### ESR\_ELx.ISV can be set incorrectly for an external abort on translation table walk

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. 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.

## 2423048

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

On exit from Debug state, PSTATE.SS is set according to DSPSR.SS and DSPSR.M.

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum has no workaround.

## 2446528

### PMU STALL\_SLOT\_BACKEND and STALL\_SLOT\_FRONTEND events count incorrectly

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

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

- 0x3D, STALL\_SLOT\_BACKEND, no operation sent for execution on a slot due to the backend
- 0x3E, STALL\_SLOT\_FRONTEND, no operation sent for execution on a slot due to the frontend

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

- 0x3D, STALL\_SLOT\_BACKEND
- 0x3E, STALL\_SLOT\_FRONTEND

#### Implications

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

#### Workaround

This erratum has no workaround.



## 2699191

### Incorrect value reported for SPE PMU event SAMPLE\_FEED

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

If the above conditions are met, then the SAMPLE\_FEED event may not be incremented.

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

#### Workaround

There is no workaround.

## 2699197

### Reads of DISR\_EL1 incorrectly return 0s while in Debug State

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

When the Processing Element (PE) is in Debug State, reads of DISR\_EL1 from EL1 or EL2 with SCR\_EL3.EA=0x1 will incorrectly return 0s.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The PE is executing in Debug State at EL1 or EL2, with SCR\_EL3.EA=0x1.
2. The PE executes an MRS to DISR\_EL1.

#### Implications

If the above conditions are met, then the read of DISR\_EL1 will incorrectly return 0s.

#### Workaround

No workaround is expected to be required.

## 2699760

### Incorrect read value for Performance Monitors Control Register

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

The Performance Monitors Control Register (PMCR\_ELO) and the External Performance Monitor Control Register (PMCR) might return an incorrect read value for the X field.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Software writes a nonzero value to the PMCR\_ELO.X, or debugger writes a nonzero value to the PMCR.X
2. Software reads the PMCR\_ELO register, or debugger reads the PMCR register

#### Implications

The PMCR\_EL1.X or PMCR.X field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

#### Workaround

This erratum has no workaround.

## 2708633

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. 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 UNDEFINED instruction.

#### Workaround

There is no workaround.

## 2712563

### Incorrect read value for Performance Monitors Configuration Register EX field

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

The Performance Monitors Configuration Register (PMCFGR) might return an incorrect read value for the EX field.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs when the software reads the PMCFGR register.

#### Implications

The PMCFGR.EX field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

#### Workaround

This erratum has no workaround.

## 2764409

### Incorrect value reported for SPE PMU event 0x4000 SAMPLE\_POP

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

Under certain conditions the SAMPLE\_POP PMU event 0x4000 might continue to count after SPE profiling has been disabled.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. *Statistical Profiling Extension* (SPE) sampling is enabled.
2. *Performance Monitoring Unit* (PMU) event counting is enabled.
3. SPE buffer is disabled, either directly by software, or indirectly via assertion of PMBIRQ, or by entry into Debug state.

#### Implications

If the previous conditions are met, then the SAMPLE\_POP event might reflect an overcounted value. The impact of this erratum is expected to be very minor for actual use cases, as SPE sampling analysis is typically performed independently from PMU event counting.

#### Workaround

If a workaround is desired, then minimization of potential overcounting of the SAMPLE\_POP event can be realized via software disable of any PMU SAMPLE\_POP event counters whenever SPE is disabled, and also upon the servicing of a PMBIRQ interrupt. For profiling of ELO workloads, software can further reduce exposure to overcounting by configuring the counter to not count at Exception levels of EL1 or higher.

## 2817022

### PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, and r1p2. Open.

#### Description

Under certain conditions, the *Processing Element* (PE) might fail to report multiple uncorrectable *Error Correction Code* (ECC) errors that occur in the L1 data cache tag RAM.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

1. The PE detects and reports an uncorrectable ECC error in the L1 data cache tag RAM.
2. The PE detects a second uncorrectable ECC error in the L1 data cache tag RAM and an uncorrectable ECC error in the L1 data cache data RAM.

#### Implications

If the previous conditions are met, then the PE might fail to report the second uncorrectable ECC error in the L1 data cache tag RAM and the address recorded in `ERR0ADDR` might have an incorrect value. The ECC error occurring in the L1 data cache data RAM is reported correctly.

#### Workaround

No workaround is necessary. This erratum represents a condition where multiple uncorrectable ECC errors occur in a short period of time. While the PE does not report the errors correctly, ECC still provides a valuable mechanism for error detection and correction.

# 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 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.



The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) 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 [0x0y] 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.