



## Arm Neoverse N2 (MP128)

### Software Developer Errata Notice

Date of issue: October 01, 2024

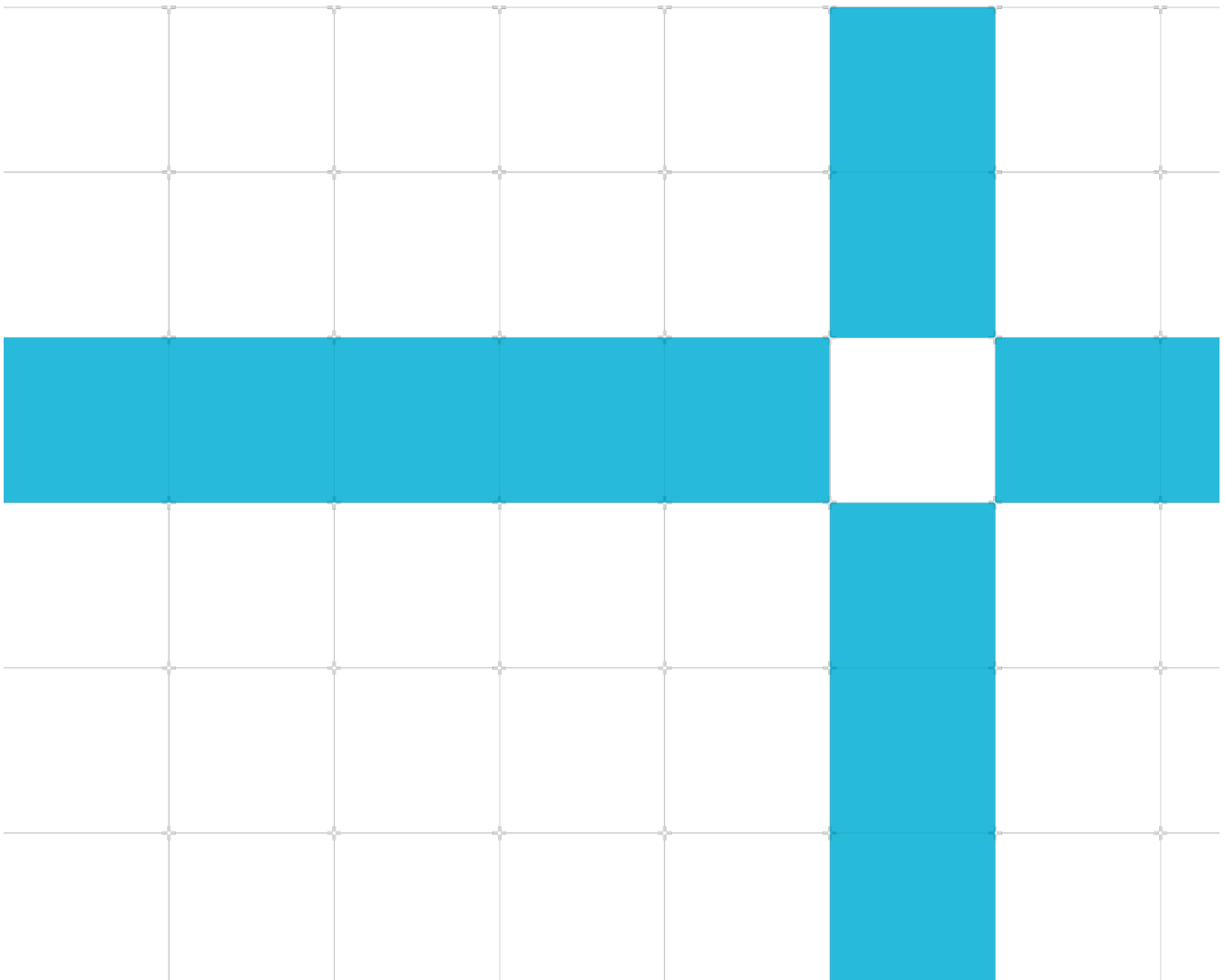
Non-Confidential

Document version: 18.0

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

Document ID: SDEN-1982442

This document contains all known errata since the r0p3 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\_1982442\_18.0\_en) was issued on October 01, 2024.

There might be a later issue at <http://developer.arm.com/documentation/SDEN-1982442>

## 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 Neoverse N2 (MP128), 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>rOp3 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>22</b>
<b>Errata descriptions</b>	<b>31</b>
Category A	31
2001293 Fault info captured in FAR and ESR registers for LDP 64-bit variant might be incorrect	31
Category A (rare)	32
Category B	33
2002655 Embedded Trace of WFI or WFE instructions might corrupt PE architectural state	33
2009478 RAS errors during core power down might cause a deadlock	35
2017090 TRBE and SPE writes to MTE tagged pages might not report external aborts	37
2025414 Streaming STG and STG2 performance lower than expected with TCF=NONE	38
2036776 Reads of ERRIDR_EL1.NUM return an incorrect value	39
2048466 Utility Bus register accesses to reserved addresses of PE might hang	40
2067956 Loss of MTE tag coherency with L2 cache tag ECC errors	41
2067961 The trace data is not flushed completely during a TSB instruction executed in prohibited region	42
2138953 Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock	43
2138956 Executing a WFI or WFE instruction after a STREX instruction might result in a deadlock under specific conditions	44
2138958 Execution of ST2G instructions in close proximity might cause loss of MTE allocation tag data	46
2139208 Trace data might get overwritten in TRBE FILL mode	47
2180384 Enabling TRBE might cause a data write to a page with the wrong ASID when owning Exception level is EL1	48
2189731 The CPP instruction will apply to an incorrect EL context	50
2242400 PDP deadlock due to CMP/CMN + B.AL/B.NV fusion	51
2242415 A CFP instruction might not invalidate the correct resources	52

2253138	TRBE might cause a data write to an out-of-range address which is not reserved for TRBE	53
2280757	A CFP instruction might execute with incorrect upper ASID or VMID bits	54
2326639	Denied power down request might prevent completion of future power down request	55
2346952	Static and dynamic TXREQ limiting might cause deadlock	56
2376738	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	57
2388448	A continuous stream of incoming DVM syncs may cause TRBE to prevent the CPU from forward progressing	58
2388450	Translation table walk folding into an L1 prefetch might cause data corruption	59
2728475	The core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back	60
2743014	Page crossing access that generates an MMU fault on the second page could result in a livelock	62
2743089	The core might deadlock during powerdown sequence	63
2779511	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation.	64
3003140	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly	65
3022731	TRBE might write to pages which lack write permission at Stage-1 or Stage-2	66
3031178	SPE might write to pages which lack write permission at Stage-1 or Stage-2	68
3099213	PE might execute instructions consistent with previous context-synchronized state when SCR_EL3.EEL2 is changed	70
3324339	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	72
3696250	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	73
3701773	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS	75
Category B (rare)		77
2340933	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment	77
2986650	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	78
Category C		80
1982438	Noncompliance with prioritization of Exception Catch debug events	80
1982439	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	82

1982440	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB	83
2002685	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL	84
2009482	Execution of STG instructions in close proximity might incorrectly write MTE Allocation Tag to memory more than once	85
2025410	DSB might not guarantee completion of direct reads of L2 cache memories	86
2033429	Corrupted register state results from executing specific form of SEL instruction followed by SVE AESMC or AESIMC instruction	87
2048467	CPU might fetch incorrect instruction from a page programmed as non-cacheable in stage-1 translation and as device memory in stage-2 translation	88
2052733	ER1PFGCTL.MV is improperly implemented as Read/Write	89
2067959	External aborts for streaming writes to MTE tagged pages may report multiple errors	90
2067960	Trace data lost during collection stop in TRBE	91
2067962	L3D_CACHE_ALLOC PMU inaccurate when using WriteEvictOrEvict transactions	92
2072986	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	93
2138952	External APB write to a register located at offset 0x084 might incorrectly issue a write to External Debug Instruction Transfer Register	95
2138954	Incorrect Fault Status code reported for predicated SVE op	97
2138955	Tag check fail might not be reported for an unaligned predicated SVE store	98
2138959	OSECCELR_EL1/EDECCR is incorrectly included in the Warm Reset domain	99
2138960	Extra A-sync packet might get written to Trace Buffer in Trace prohibited region	100
2139204	Speculative access to a recently unmapped physical address previously containing page tables might occur	101
2139205	L1D_CACHE_INVALID and L2D_CACHE_INVALID PMU events fail to increment for SnpPreferUnique and SnpPreferUniqueFwd	102
2139206	MPAM value associated with instruction fetch might be incorrect	103
2139207	Data abort on SVE first fault load might be routed to incorrect Exception level	104
2149120	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely	105
2153906	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes	107
2189737	PMU Event MEM_ACCESS_CHECKED_WR, 0x4026 counts incorrectly and MEM_ACC_CHECKED 0x4024 might be incorrect	108
2189738	Direct access of L2 data RAMs using RAMINDEX returns incomplete data	109
2189739	PMU_HOVFS event not always exported when self-hosted trace is disabled	110
2218242	Some SVE PMU events count incorrectly	111

2218243	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect	113
2240288	Incorrect sampling of SPE events "Partial predicate" and "Empty predicate" for SVE instruction with no vector operands	114
2242397	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering	115
2242404	64 bit source SVE PMULLB/T not considered Cryptography instruction	116
2242416	An SError might not be reported for an atomic store that encounters data poison	117
2253154	ESR_ELx contents for a Data Abort exception might be incorrect when an L1D tag double bit error is encountered	118
2276004	Incorrect Fault Status Code reported on a SPE buffer translation fault	119
2280302	Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work	120
2280397	PMU L1D_CACHE_REFILL_OUTER is inaccurate	121
2307062	Reads of DISR_EL1 incorrectly return 0s while in Debug State	122
2307063	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state	123
2307064	ELR_ELx[63:48] might hold incorrect value when PE disables address translation	124
2307065	TRBE might use incorrect Cacheability attributes for TRBE data when address translation is disabled	125
2307066	L1 MTE Tag poison is not cleared	126
2341660	ESR_ELx contents for a Data Abort exception might be incorrect when a data double bit error or external abort is encountered	127
2341661	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk	128
2341669	SPE, PMU event for full/partial/empty/not full predicate might be incorrect for some SVE instructions	129
2341671	Lower priority exception might be reported when abort condition is detected at both stages of translation	131
2346744	CSSELR_EL1.TnD is RAZ/WI when CSSELR_EL1.InD == 0x1	132
2346745	L2 tag RAM double-bit ECC error might lead to the PE not responding to a forwarding snoop	133
2393116	L1 Data poison is not cleared by a store	134
2446315	Software-step not done after exit from Debug state with an illegal value in DSPSR	135
2446526	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly	136
2639704	STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly	137
2648408	ERXPFgcdN_EL1 register is incorrectly written on Warm reset	138
2648410	Incorrect read value for Performance Monitors Control Register	139
2664787	Incorrect value reported for SPE PMU event SAMPLE_FEED	140
2675383	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect	141

2696764	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data	142
2728476	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering	143
2728477	Incorrect read value for Performance Monitors Configuration Register EX field	144
2736656	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFE state	145
2738454	PMU events STALL_SLOT_FRONTEND and STALL_SLOT count incorrectly	146
2755354	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	147
2794918	DGH instruction doesn't execute correctly	148
2803644	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions	149
2814366	ECC errors in MTE allocation tags may lead to silent data corruption in tag values	151
2814415	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	152
2814419	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	153
2820250	TRBE buffer write translation out of context may have incorrect memory attributes	154
2914113	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption	155
2933583	L2D_CACHE_WB_CLEAN overcounts	156
2973112	IRG instructions might produce the wrong tag when GCR_EL1.RRND=0x0.	157
3070040	TagMatch responses with error indication do not generate a SError abort	158
3604862	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative	159
3605043	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed	160
3627241	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability	162
3633461	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception	163
3640937	SPE operation type is corrupted under certain conditions	164
3694436	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled	165
3694458	FFR might not capture the lowest faulting memory element	166
3700127	PE might fail to log a RAS error for L2 data RAM ECC errors	167
3705908	PMU events are mis-categorized by not considering the effect of "Taken locally"	168
<b>Proprietary notice</b>		169
<b>Product and document information</b>		171

<b>Product status</b>	<b>171</b>
<b>Product completeness status</b>	<b>171</b>
<b>Product revision status</b>	<b>171</b>

## 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]	2001293 Fault info captured in FAR and ESR registers for LDP 64-bit variant could be incorrect
REVIDR_EL1[1]	2346952 Static and dynamic TXREQ limiting might cause deadlock
REVIDR_EL1[2]	2346952 Static and dynamic TXREQ limiting might cause deadlock

Note that there is no change to the MIDR\_EL1 which remains at rOp0. Software will identify this release through the combination of MIDR\_EL1 and REVIDR\_EL1.

## rOp3 implementation fixes

Note the following errata might be fixed in some implementations of rOp3. 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]	2639704 STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly
REVIDR_EL1[1]	2986650 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

Note that there is no change to the MIDR\_EL1 which remains at rOp3. 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.

## October 01, 2024: Changes in document version v18.0

ID	Status	Area	Category	Summary
<a href="#">3696250</a>	New	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock
<a href="#">3701773</a>	New	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS
<a href="#">3604862</a>	New	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative
<a href="#">3605043</a>	New	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed
<a href="#">3627241</a>	New	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability
<a href="#">3633461</a>	New	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception
<a href="#">3640937</a>	New	Programmer	Category C	SPE operation type is corrupted under certain conditions
<a href="#">3694436</a>	New	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled
<a href="#">3694458</a>	New	Programmer	Category C	FFR might not capture the lowest faulting memory element
<a href="#">3700127</a>	New	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors
<a href="#">3705908</a>	New	Programmer	Category C	PMU events are mis-categorized by not considering the effect of Taken locally

## April 30, 2024: Changes in document version v17.0

ID	Status	Area	Category	Summary
<a href="#">3324339</a>	New	Programmer	Category B	MSR PSTATE.SBS to 0 is not fully self-synchronizing

## December 15, 2023: Changes in document version v16.0

ID	Status	Area	Category	Summary
<a href="#">3022731</a>	Updated	Programmer	Category B	TRBE might write to pages which lack write permission at Stage-1 or Stage-2
<a href="#">3099213</a>	New	Programmer	Category B	PE might execute instructions consistent with previous context-synchronized state when SCR_EL3.EEL2 is changed
<a href="#">3070040</a>	New	Programmer	Category C	TagMatch responses with error indication do not generate a SError abort

## September 01, 2023: Changes in document version v15.0

ID	Status	Area	Category	Summary
<a href="#">3003140</a>	New	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly
<a href="#">3022731</a>	New	Programmer	Category B	TRBE might write to pages which lack write permission at Stage-1 or Stage-2
<a href="#">3031178</a>	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
<a href="#">2794918</a>	New	Programmer	Category C	DGH instruction doesn't execute correctly
<a href="#">2914113</a>	New	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption
<a href="#">2933583</a>	New	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts
<a href="#">2973112</a>	New	Programmer	Category C	IRG instructions might produce the wrong tag when GCR_EL1.RRND=0x0

## July 25, 2023: Changes in document version v14.0

ID	Status	Area	Category	Summary
<a href="#">2986650</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
<a href="#">2639704</a>	Updated	Programmer	Category C	STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly

## February 22, 2023: Changes in document version v13.0

ID	Status	Area	Category	Summary
<a href="#">2803644</a>	New	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions
<a href="#">2814366</a>	New	Programmer	Category C	ECC errors in MTE allocation tags may lead to silent data corruption in tag values
<a href="#">2814415</a>	New	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled
<a href="#">2814419</a>	New	Programmer	Category C	PE may fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM
<a href="#">2820250</a>	New	Programmer	Category C	TRBE buffer write translation out of context may have incorrect memory attributes

## November 01, 2022: Changes in document version v12.0

ID	Status	Area	Category	Summary
<a href="#">2743014</a>	New	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock
<a href="#">2743089</a>	New	Programmer	Category B	The core might deadlock during powerdown sequence
<a href="#">2779511</a>	New	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation
<a href="#">2639704</a>	New	Programmer	Category C	STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly
<a href="#">2736656</a>	New	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFE state
<a href="#">2738454</a>	New	Programmer	Category C	PMU events STALL_SLOT_FRONTEND and STALL_SLOT count incorrectly
<a href="#">2755354</a>	New	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

## September 09, 2022: Changes in document version v11.0

ID	Status	Area	Category	Summary
<a href="#">2346952</a>	New	Programmer	Category B	Static and dynamic TXREQ limiting might cause deadlock
<a href="#">2376738</a>	Updated	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
<a href="#">2728475</a>	New	Programmer	Category B	CPU might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back
<a href="#">2664787</a>	New	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED
<a href="#">2675383</a>	New	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect
<a href="#">2696764</a>	New	Programmer	Category C	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data
<a href="#">2728476</a>	New	Programmer	Category C	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering
<a href="#">2728477</a>	New	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field

## May 26, 2022: Changes in document version v10.0

ID	Status	Area	Category	Summary
<a href="#">2340933</a>	Updated	Programmer	Category B (rare)	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment
<a href="#">2446315</a>	New	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR
<a href="#">2446526</a>	New	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly
<a href="#">2648408</a>	New	Programmer	Category C	ERXPFCDN_EL1 register is incorrectly written on Warm reset
<a href="#">2648410</a>	New	Programmer	Category C	Incorrect read value for Performance Monitors Control Register

## December 16, 2021: Changes in document version v9.0

ID	Status	Area	Category	Summary
<a href="#">2001293</a>	Updated	Programmer	Category A	Fault info captured in FAR and ESR registers for LDP 64-bit variant could be incorrect
<a href="#">2002655</a>	Updated	Programmer	Category B	Embedded Trace of WFI or WFE instructions might corrupt PE architectural state
<a href="#">2009478</a>	Updated	Programmer	Category B	RAS errors during core power down might cause a deadlock
<a href="#">2017090</a>	Updated	Programmer	Category B	TRBE and SPE writes to MTE tagged pages might not report external aborts
<a href="#">2025414</a>	Updated	Programmer	Category B	Streaming STG and STG2 performance lower than expected with TCF=NONE
<a href="#">2036776</a>	Updated	Programmer	Category B	Reads of ERRIDR_EL1.NUM return an incorrect value
<a href="#">2048466</a>	Updated	Programmer	Category B	Utility Bus register accesses to reserved addresses of PE might hang
<a href="#">2067956</a>	Updated	Programmer	Category B	Loss of MTE tag coherency with L2 cache tag ECC errors
<a href="#">2067961</a>	Updated	Programmer	Category B	The trace data is not flushed completely during a TSB instruction executed in prohibited region
<a href="#">2138956</a>	Updated	Programmer	Category B	Executing a WFI or WFE instruction after a STREX instruction might result in a deadlock under specific conditions
<a href="#">2138958</a>	Updated	Programmer	Category B	Execution of ST2G instructions in close proximity might cause loss of MTE allocation tag data
<a href="#">2139208</a>	Updated	Programmer	Category B	Trace data might get overwritten in TRBE FILL mode
<a href="#">2180384</a>	Updated	Programmer	Category B	Enabling TRBE might cause a data write to a page with the wrong ASID when owning Exception level is EL1
<a href="#">2189731</a>	Updated	Programmer	Category B	The CPP instruction will apply to an incorrect EL context
<a href="#">2242400</a>	Updated	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion
<a href="#">2242415</a>	Updated	Programmer	Category B	A CFP instruction might not invalidate the correct resources
<a href="#">2253138</a>	Updated	Programmer	Category B	TRBE might cause a data write to an out-of-range address which is not reserved for TRBE
<a href="#">2280757</a>	Updated	Programmer	Category B	A CFP instruction might execute with incorrect upper ASID or VMID bits
<a href="#">2326639</a>	Updated	Programmer	Category B	Denied power down request might prevent completion of future power down request
<a href="#">2376738</a>	New	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
<a href="#">2388448</a>	New	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the CPU from forward progressing
<a href="#">2388450</a>	New	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption
<a href="#">2340933</a>	New	Programmer	Category B (rare)	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment

ID	Status	Area	Category	Summary
<a href="#">2002685</a>	Updated	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL
<a href="#">2009482</a>	Updated	Programmer	Category C	Execution of STG instructions in close proximity might incorrectly write MTE Allocation Tag to memory more than once
<a href="#">2025410</a>	Updated	Programmer	Category C	DSB might not guarantee completion of direct reads of L2 cache memories
<a href="#">2033429</a>	Updated	Programmer	Category C	Corrupted register state results from executing specific form of SEL instruction followed by SVE AESMC or AESIMC instruction
<a href="#">2048467</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="#">2052733</a>	Updated	Programmer	Category C	ER1PFGCTL.MV is improperly implemented as Read/Write
<a href="#">2067959</a>	Updated	Programmer	Category C	External aborts for streaming writes to MTE tagged pages may report multiple errors
<a href="#">2067960</a>	Updated	Programmer	Category C	Trace data lost during collection stop in TRBE
<a href="#">2067962</a>	Updated	Programmer	Category C	L3D_CACHE_ALLOC PMU inaccurate when using WriteEvictOrEvict transactions
<a href="#">2072986</a>	Updated	Programmer	Category C	An execution of MSR instruction might not update the destination register correctly when an external debugger initiates APB write operation to update debug registers
<a href="#">2138952</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="#">2138954</a>	Updated	Programmer	Category C	Incorrect Fault Status code reported for predicated SVE op
<a href="#">2138955</a>	Updated	Programmer	Category C	Tag check fail might not be reported for an unaligned predicated SVE store
<a href="#">2138959</a>	Updated	Programmer	Category C	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain
<a href="#">2138960</a>	Updated	Programmer	Category C	Extra A-sync packet might get written to Trace Buffer in Trace prohibited region
<a href="#">2139204</a>	Updated	Programmer	Category C	Speculative access to a recently unmapped physical address previously containing page tables might occur
<a href="#">2139205</a>	Updated	Programmer	Category C	L1D_CACHE_INVALID and L2D_CACHE_INVALID PMU events fail to increment for SnpPreferUnique and SnpPreferUniqueFwd
<a href="#">2139207</a>	Updated	Programmer	Category C	Data abort on SVE first fault load might be routed to incorrect Exception level
<a href="#">2149120</a>	Updated	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely
<a href="#">2153906</a>	Updated	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes
<a href="#">2189737</a>	Updated	Programmer	Category C	PMU Event MEM_ACCESS_CHECKED_WR, 0x4026 counts incorrectly and MEM_ACC_CHECKED 0x4024 might be incorrect

ID	Status	Area	Category	Summary
<a href="#">2189738</a>	Updated	Programmer	Category C	Direct access of L2 data RAMs using RAMINDEX returns incomplete data
<a href="#">2189739</a>	Updated	Programmer	Category C	PMU_HOVFS event not always exported when self-hosted trace is disabled
<a href="#">2218242</a>	Updated	Programmer	Category C	Some SVE PMU events count incorrectly
<a href="#">2218243</a>	Updated	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect
<a href="#">2240288</a>	Updated	Programmer	Category C	Incorrect sampling of SPE events "Partial predicate" and "Empty predicate" for SVE instruction with no vector operands
<a href="#">2242397</a>	Updated	Programmer	Category C	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering
<a href="#">2242404</a>	Updated	Programmer	Category C	64 bit source SVE PMULLB/T not considered Cryptography instruction
<a href="#">2242416</a>	Updated	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison
<a href="#">2253154</a>	Updated	Programmer	Category C	ESR_ELx contents for a Data Abort exception might be incorrect when an L1D tag double bit error is encountered
<a href="#">2276004</a>	New	Programmer	Category C	Incorrect Fault Status Code reported on a SPE buffer translation fault
<a href="#">2280302</a>	Updated	Programmer	Category C	Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work
<a href="#">2280397</a>	New	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate
<a href="#">2307062</a>	Updated	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State
<a href="#">2307063</a>	Updated	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state
<a href="#">2307064</a>	Updated	Programmer	Category C	ELR_ELx[63:48] might hold incorrect value when PE disables address translation
<a href="#">2307065</a>	Updated	Programmer	Category C	TRBE might use incorrect Cacheability attributes for TRBE data when address translation is disabled
<a href="#">2307066</a>	Updated	Programmer	Category C	L1 MTE Tag poison is not cleared
<a href="#">2341660</a>	New	Programmer	Category C	ESR_ELx contents for a Data Abort exception might be incorrect when a data double bit error or external abort is encountered
<a href="#">2341661</a>	New	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk
<a href="#">2341669</a>	New	Programmer	Category C	SPE, PMU event for full/partial/empty/not full predicate might be incorrect for some SVE instructions
<a href="#">2341671</a>	New	Programmer	Category C	Lower priority exception might be reported when abort condition is detected at both stages of translation
<a href="#">2346744</a>	New	Programmer	Category C	CSSELR_EL1.TnD is RAZ/WI when CSSELR_EL1.InD == 0x1
<a href="#">2346745</a>	New	Programmer	Category C	L2 tag RAM double-bit ECC error might lead to the PE not responding to a forwarding snoop

ID	Status	Area	Category	Summary
<a href="#">2393116</a>	New	Programmer	Category C	L1 Data poison is not cleared by a store

## October 08, 2021: Changes in document version v8.0

ID	Status	Area	Category	Summary
<a href="#">2009478</a>	Updated	Programmer	Category B	RAS errors during core power down might cause a deadlock
<a href="#">2242415</a>	Updated	Programmer	Category B	A CFP instruction might not invalidate the correct resources
<a href="#">2280757</a>	New	Programmer	Category B	A CFP instruction might execute with incorrect upper ASID or VMID bits
<a href="#">2326639</a>	New	Programmer	Category B	Denied power down request might prevent completion of future power down request
<a href="#">2149120</a>	Updated	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely
<a href="#">2240288</a>	New	Programmer	Category C	Incorrect sampling of SPE events "Partial predicate" and "Empty predicate" for SVE instruction with no vector operands
<a href="#">2280302</a>	New	Programmer	Category C	Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work
<a href="#">2307062</a>	New	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State
<a href="#">2307063</a>	New	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at EL0 in Debug state
<a href="#">2307064</a>	New	Programmer	Category C	ELR_ELx[63:48] might hold incorrect value when PE disables address translation
<a href="#">2307065</a>	New	Programmer	Category C	TRBE might use incorrect Cacheability attributes for TRBE data when address translation is disabled
<a href="#">2307066</a>	New	Programmer	Category C	L1 MTE Tag poison is not cleared

## July 16, 2021: Changes in document version v7.0

ID	Status	Area	Category	Summary
<a href="#">2180384</a>	New	Programmer	Category B	Enabling TRBE might cause a data write to a page with the wrong ASID when owning Exception level is EL1
<a href="#">2242400</a>	New	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion
<a href="#">2242415</a>	New	Programmer	Category B	A CFP instruction might not invalidate the correct resources
<a href="#">2253138</a>	New	Programmer	Category B	TRBE might cause a data write to an out-of-range address which is not reserved for TRBE
<a href="#">2242397</a>	New	Programmer	Category C	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering
<a href="#">2242404</a>	New	Programmer	Category C	64 bit source SVE PMULLB/T not considered Cryptography instruction
<a href="#">2242416</a>	New	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison
<a href="#">2253154</a>	New	Programmer	Category C	ESR_ELx contents for a Data Abort exception might be incorrect when an L1D tag double bit error is encountered

## June 08, 2021: Changes in document version v6.0

ID	Status	Area	Category	Summary
<a href="#">2002655</a>	Updated	Programmer	Category B	Embedded Trace of WFI or WFE instructions might corrupt PE architectural state
<a href="#">2189731</a>	New	Programmer	Category B	The CPP instruction will apply to an incorrect EL context
<a href="#">2149120</a>	New	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely
<a href="#">2153906</a>	New	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes
<a href="#">2189737</a>	New	Programmer	Category C	PMU Event MEM_ACCESS_CHECKED_WR, 0x4026 counts incorrectly and MEM_ACC_CHECKED 0x4024 might be incorrect
<a href="#">2189738</a>	New	Programmer	Category C	Direct access of L2 data RAMs using RAMINDEX returns incomplete data
<a href="#">2189739</a>	New	Programmer	Category C	PMU_HOVFS event not always exported when self-hosted trace is disabled
<a href="#">2218242</a>	New	Programmer	Category C	Some SVE PMU events count incorrectly
<a href="#">2218243</a>	New	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect

## April 14, 2021: Changes in document version v5.0

ID	Status	Area	Category	Summary
<a href="#">2138953</a>	New	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock
<a href="#">2138956</a>	New	Programmer	Category B	Executing a WFI or WFE instruction after a STREX instruction might result in a deadlock under specific conditions
<a href="#">2138958</a>	New	Programmer	Category B	Execution of ST2G instructions in close proximity might cause loss of MTE allocation tag data
<a href="#">2139208</a>	New	Programmer	Category B	Trace data might get overwritten in TRBE FILL mode
<a href="#">2138952</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="#">2138954</a>	New	Programmer	Category C	Incorrect Fault Status code reported for predicated SVE op
<a href="#">2138955</a>	New	Programmer	Category C	Tag check fail might not be reported for an unaligned predicated SVE store
<a href="#">2138959</a>	New	Programmer	Category C	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain
<a href="#">2138960</a>	New	Programmer	Category C	Extra A-sync packet might get written to Trace Buffer in Trace prohibited region
<a href="#">2139204</a>	New	Programmer	Category C	Speculative access to a recently unmapped physical address previously containing page tables might occur
<a href="#">2139205</a>	New	Programmer	Category C	L1D_CACHE_INVALID and L2D_CACHE_INVALID PMU events fail to increment for SnpPreferUnique and SnpPreferUniqueFwd
<a href="#">2139206</a>	New	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect
<a href="#">2139207</a>	New	Programmer	Category C	Data abort on SVE first fault load might be routed to incorrect Exception level

**February 02, 2021: Changes in document version v4.0**

ID	Status	Area	Category	Summary
<a href="#">2036776</a>	New	Programmer	Category B	Reads of ERRIDR_EL1.NUM return an incorrect value
<a href="#">2048466</a>	New	Programmer	Category B	Utility Bus register accesses to reserved addresses of PE might hang
<a href="#">2067956</a>	New	Programmer	Category B	Loss of MTE tag coherency with L2 cache tag ECC errors
<a href="#">2067961</a>	New	Programmer	Category B	The trace data is not flushed completely during a TSB instruction executed in prohibited region
<a href="#">2048467</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="#">2052733</a>	New	Programmer	Category C	ER1PFGCTL.MV is improperly implemented as Read/Write
<a href="#">2067959</a>	New	Programmer	Category C	External aborts for streaming writes to MTE tagged pages may report multiple errors
<a href="#">2067960</a>	New	Programmer	Category C	Trace data lost during collection stop in TRBE
<a href="#">2067962</a>	New	Programmer	Category C	L3D_CACHE_ALLOC PMU inaccurate when using WriteEvictOrEvict transactions
<a href="#">2072986</a>	New	Programmer	Category C	An execution of MSR instruction might not update the destination register correctly when an external debugger initiates APB write operation to update debug registers

**December 08, 2020: Changes in document version v3.0**

ID	Status	Area	Category	Summary
<a href="#">2009478</a>	New	Programmer	Category B	RAS errors during core power down might cause a deadlock
<a href="#">2017090</a>	New	Programmer	Category B	TRBE and SPE writes to MTE tagged pages might not report external aborts
<a href="#">2025414</a>	New	Programmer	Category B	Streaming STG and STG2 performance lower than expected with TCF=NONE
<a href="#">2025410</a>	New	Programmer	Category C	DSB might not guarantee completion of direct reads of L2 cache memories
<a href="#">2033429</a>	New	Programmer	Category C	Corrupted register state results from executing specific form of SEL instruction followed by SVE AESMC or AESIMC instruction

**November 06, 2020: Changes in document version v2.0**

ID	Status	Area	Category	Summary
<a href="#">2001293</a>	New	Programmer	Category A	Fault info captured in FAR and ESR registers for LDP 64-bit variant could be incorrect
<a href="#">2002655</a>	New	Programmer	Category B	Embedded Trace of WFI or WFE instructions might corrupt PE architectural state
<a href="#">2002685</a>	New	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ElX.IESB set in the current EL
<a href="#">2009482</a>	New	Programmer	Category C	Execution of STG instructions in close proximity might incorrectly write MTE Allocation Tag to memory more than once

## October 16, 2020: Changes in document version v1.0

ID	Status	Area	Category	Summary
<a href="#">1982438</a>	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events
<a href="#">1982439</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 page table entry which has been modified
<a href="#">1982440</a>	New	Programmer	Category C	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB

# 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="#">2001293</a>	Programmer	Category A	Fault info captured in FAR and ESR registers for LDP 64-bit variant could be incorrect	r0p0	r0p1
<a href="#">2002655</a>	Programmer	Category B	Embedded Trace of WFI or WFE instructions might corrupt PE architectural state	r0p0	r0p1
<a href="#">2009478</a>	Programmer	Category B	RAS errors during core power down might cause a deadlock	r0p0	r0p1
<a href="#">2017090</a>	Programmer	Category B	TRBE and SPE writes to MTE tagged pages might not report external aborts	r0p0	r0p1
<a href="#">2025414</a>	Programmer	Category B	Streaming STG and STG2 performance lower than expected with TCF=NONE	r0p0	r0p1
<a href="#">2036776</a>	Programmer	Category B	Reads of ERRIDR_EL1.NUM return an incorrect value	r0p0	r0p1
<a href="#">2048466</a>	Programmer	Category B	Utility Bus register accesses to reserved addresses of PE might hang	r0p0	r0p1
<a href="#">2067956</a>	Programmer	Category B	Loss of MTE tag coherency with L2 cache tag ECC errors	r0p0	r0p1
<a href="#">2067961</a>	Programmer	Category B	The trace data is not flushed completely during a TSB instruction executed in prohibited region	r0p0	r0p1
<a href="#">2138953</a>	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2138956</a>	Programmer	Category B	Executing a WFI or WFE instruction after a STREX instruction might result in a deadlock under specific conditions	r0p0	r0p1
<a href="#">2138958</a>	Programmer	Category B	Execution of ST2G instructions in close proximity might cause loss of MTE allocation tag data	r0p0	r0p1
<a href="#">2139208</a>	Programmer	Category B	Trace data might get overwritten in TRBE FILL mode	r0p0	r0p1
<a href="#">2180384</a>	Programmer	Category B	Enabling TRBE might cause a data write to a page with the wrong ASID when owning Exception level is EL1	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2189731</a>	Programmer	Category B	The CPP instruction will apply to an incorrect EL context	r0p0	r0p1
<a href="#">2242400</a>	Programmer	Category B	PDP deadlock due to CMP/CMN + B.AL/B.NV fusion	r0p0	r0p1
<a href="#">2242415</a>	Programmer	Category B	A CFP instruction might not invalidate the correct resources	r0p0	r0p1
<a href="#">2253138</a>	Programmer	Category B	TRBE might cause a data write to an out-of-range address which is not reserved for TRBE	r0p0	r0p1
<a href="#">2280757</a>	Programmer	Category B	A CFP instruction might execute with incorrect upper ASID or VMID bits	r0p0	r0p1
<a href="#">2326639</a>	Programmer	Category B	Denied power down request might prevent completion of future power down request	r0p0	r0p1
<a href="#">2346952</a>	Programmer	Category B	Static and dynamic TXREQ limiting might cause deadlock	r0p0, r0p1, r0p2	r0p3
<a href="#">2376738</a>	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2388448</a>	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the CPU from forward progressing	r0p0	r0p1
<a href="#">2388450</a>	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption	r0p0	r0p1
<a href="#">2728475</a>	Programmer	Category B	CPU might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back	r0p0, r0p1, r0p2	r0p3
<a href="#">2743014</a>	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock	r0p0, r0p1, r0p2	r0p3
<a href="#">2743089</a>	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r0p1, r0p2	r0p3
<a href="#">2779511</a>	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation	r0p0, r0p1, r0p2	r0p3
<a href="#">3003140</a>	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly	r0p0, r0p1, r0p2, r0p3	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">3022731</a>	Programmer	Category B	TRBE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0	r0p1
<a href="#">3031178</a>	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3099213</a>	Programmer	Category B	PE might execute instructions consistent with previous context-synchronized state when SCR_EL3.EEL2 is changed	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3324339</a>	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3696250</a>	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3701773</a>	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2340933</a>	Programmer	Category B (rare)	AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment	r0p0	r0p1
<a href="#">2986650</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, r0p1, r0p2, r0p3	Open
<a href="#">1982438</a>	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">1982439</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 page table entry which has been modified	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">1982440</a>	Programmer	Category C	IDATAN_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2002685</a>	Programmer	Category C	DRPS might not execute correctly in Debug state with SCTLR_ELx.IESB set in the current EL	r0p0	r0p1
<a href="#">2009482</a>	Programmer	Category C	Execution of STG instructions in close proximity might incorrectly write MTE Allocation Tag to memory more than once	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2025410</a>	Programmer	Category C	DSB might not guarantee completion of direct reads of L2 cache memories	r0p0	r0p1
<a href="#">2033429</a>	Programmer	Category C	Corrupted register state results from executing specific form of SEL instruction followed by SVE AESMC or AESIMC instruction	r0p0	r0p1
<a href="#">2048467</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	r0p1
<a href="#">2052733</a>	Programmer	Category C	ER1PFGCTL.MV is improperly implemented as Read/Write	r0p0	r0p1
<a href="#">2067959</a>	Programmer	Category C	External aborts for streaming writes to MTE tagged pages may report multiple errors	r0p0	r0p1
<a href="#">2067960</a>	Programmer	Category C	Trace data lost during collection stop in TRBE	r0p0	r0p1
<a href="#">2067962</a>	Programmer	Category C	L3D_CACHE_ALLOC PMU inaccurate when using WriteEvictOrEvict transactions	r0p0	r0p1
<a href="#">2072986</a>	Programmer	Category C	An execution of MSR instruction might not update the destination register correctly when an external debugger initiates APB write operation to update debug registers	r0p0	r0p1
<a href="#">2138952</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	r0p1
<a href="#">2138954</a>	Programmer	Category C	Incorrect Fault Status code reported for predicated SVE op	r0p0	r0p1
<a href="#">2138955</a>	Programmer	Category C	Tag check fail might not be reported for an unaligned predicated SVE store	r0p0	r0p1
<a href="#">2138959</a>	Programmer	Category C	OSECCR_EL1/EDECCR is incorrectly included in the Warm Reset domain	r0p0	r0p1
<a href="#">2138960</a>	Programmer	Category C	Extra A-sync packet might get written to Trace Buffer in Trace prohibited region	r0p0	r0p1
<a href="#">2139204</a>	Programmer	Category C	Speculative access to a recently unmapped physical address previously containing page tables might occur	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2139205</a>	Programmer	Category C	L1D_CACHE_INVALID and L2D_CACHE_INVALID PMU events fail to increment for SnpPreferUnique and SnpPreferUniqueFwd	r0p0	r0p1
<a href="#">2139206</a>	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2139207</a>	Programmer	Category C	Data abort on SVE first fault load might be routed to incorrect Exception level	r0p0	r0p1
<a href="#">2149120</a>	Programmer	Category C	A64 WFI or A64 WFE executed in Debug state suspends execution indefinitely	r0p0	r0p1
<a href="#">2153906</a>	Programmer	Category C	Collision bit in PMBSR is reported incorrectly when there are multiple errors on SPE writes	r0p0	r0p1
<a href="#">2189737</a>	Programmer	Category C	PMU Event MEM_ACCESS_CHECKED_WR, 0x4026 counts incorrectly and MEM_ACC_CHECKED 0x4024 might be incorrect	r0p0	r0p1
<a href="#">2189738</a>	Programmer	Category C	Direct access of L2 data RAMs using RAMINDEX returns incomplete data	r0p0	r0p1
<a href="#">2189739</a>	Programmer	Category C	PMU_HOVFS event not always exported when self-hosted trace is disabled	r0p0	r0p1
<a href="#">2218242</a>	Programmer	Category C	Some SVE PMU events count incorrectly	r0p0	r0p1
<a href="#">2218243</a>	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect	r0p0	r0p1
<a href="#">2240288</a>	Programmer	Category C	Incorrect sampling of SPE events "Partial predicate" and "Empty predicate" for SVE instruction with no vector operands	r0p0	r0p1
<a href="#">2242397</a>	Programmer	Category C	Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering	r0p0	r0p1
<a href="#">2242404</a>	Programmer	Category C	64 bit source SVE PMULLB/T not considered Cryptography instruction	r0p0	r0p1
<a href="#">2242416</a>	Programmer	Category C	An SError might not be reported for an atomic store that encounters data poison	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2253154</a>	Programmer	Category C	ESR_ELx contents for a Data Abort exception might be incorrect when an L1D tag double bit error is encountered	r0p0	r0p1
<a href="#">2276004</a>	Programmer	Category C	Incorrect Fault Status Code reported on a SPE buffer translation fault	r0p0	r0p1
<a href="#">2280302</a>	Programmer	Category C	Read or write from Secure EL1 for ICV_BPR1_EL1 register might not work	r0p0	r0p1
<a href="#">2280397</a>	Programmer	Category C	PMU L1D_CACHE_REFILL_OUTER is inaccurate	r0p0	r0p1
<a href="#">2307062</a>	Programmer	Category C	Reads of DISR_EL1 incorrectly return 0s while in Debug State	r0p0	r0p1
<a href="#">2307063</a>	Programmer	Category C	DRPS instruction is not treated as UNDEFINED at ELO in Debug state	r0p0	r0p1
<a href="#">2307064</a>	Programmer	Category C	ELR_ELx[63:48] might hold incorrect value when PE disables address translation	r0p0	r0p1
<a href="#">2307065</a>	Programmer	Category C	TRBE might use incorrect Cacheability attributes for TRBE data when address translation is disabled	r0p0	r0p1
<a href="#">2307066</a>	Programmer	Category C	L1 MTE Tag poison is not cleared	r0p0	r0p1
<a href="#">2341660</a>	Programmer	Category C	ESR_ELx contents for a Data Abort exception might be incorrect when a data double bit error or external abort is encountered	r0p0	r0p1
<a href="#">2341661</a>	Programmer	Category C	ESR_ELx.ISV can be set incorrectly for an external abort on translation table walk	r0p0	r0p1
<a href="#">2341669</a>	Programmer	Category C	SPE, PMU event for full/partial/empty/not full predicate might be incorrect for some SVE instructions	r0p0	r0p1
<a href="#">2341671</a>	Programmer	Category C	Lower priority exception might be reported when abort condition is detected at both stages of translation	r0p0	r0p1
<a href="#">2346744</a>	Programmer	Category C	CSSELR_EL1.TnD is RAZ/WI when CSSELR_EL1.InD == 0x1	r0p0	r0p1
<a href="#">2346745</a>	Programmer	Category C	L2 tag RAM double-bit ECC error might lead to the PE not responding to a forwarding snoop	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2393116</a>	Programmer	Category C	L1 Data poison is not cleared by a store	r0p0	r0p1
<a href="#">2446315</a>	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2446526</a>	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly	r0p0, r0p1	r0p2
<a href="#">2639704</a>	Programmer	Category C	STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly	r0p3	Open
<a href="#">2648408</a>	Programmer	Category C	ERXPFGCDN_EL1 register is incorrectly written on Warm reset	r0p0, r0p1, r0p2	r0p3
<a href="#">2648410</a>	Programmer	Category C	Incorrect read value for Performance Monitors Control Register	r0p0, r0p1, r0p2	r0p3
<a href="#">2664787</a>	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED	r0p0, r0p1, r0p2	r0p3
<a href="#">2675383</a>	Programmer	Category C	FAR_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect	r0p0, r0p1, r0p2	r0p3
<a href="#">2696764</a>	Programmer	Category C	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data	r0p0, r0p1, r0p2	r0p3
<a href="#">2728476</a>	Programmer	Category C	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2728477</a>	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field	r0p0, r0p1, r0p2	r0p3
<a href="#">2736656</a>	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFE state	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2738454</a>	Programmer	Category C	PMU events STALL_SLOT_FRONTEND and STALL_SLOT count incorrectly	r0p0, r0p1, r0p2	r0p3
<a href="#">2755354</a>	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	r0p0, r0p1, r0p2	r0p3
<a href="#">2794918</a>	Programmer	Category C	DGH instruction doesn't execute correctly	r0p0, r0p1, r0p2, r0p3	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2803644</a>	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2814366</a>	Programmer	Category C	ECC errors in MTE allocation tags may lead to silent data corruption in tag values	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2814415</a>	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2814419</a>	Programmer	Category C	PE may fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2820250</a>	Programmer	Category C	TRBE buffer write translation out of context may have incorrect memory attributes	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2914113</a>	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2933583</a>	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">2973112</a>	Programmer	Category C	IRG instructions might produce the wrong tag when GCR_EL1.RRND=0x0	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3070040</a>	Programmer	Category C	TagMatch responses with error indication do not generate a SError abort	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3604862</a>	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3605043</a>	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3627241</a>	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3633461</a>	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3640937</a>	Programmer	Category C	SPE operation type is corrupted under certain conditions	r0p0, r0p1, r0p2, r0p3	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">3694436</a>	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3694458</a>	Programmer	Category C	FFR might not capture the lowest faulting memory element	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3700127</a>	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors	r0p0, r0p1, r0p2, r0p3	Open
<a href="#">3705908</a>	Programmer	Category C	PMU events are mis-categorized by not considering the effect of Taken locally	r0p0, r0p1, r0p2, r0p3	Open

# Errata descriptions

## Category A

2001293

Fault info captured in FAR and ESR registers for LDP 64-bit variant might be incorrect

### Status

Fault Type: Programmer Category A

Fault Status: Present in r0p0. Fixed in r0p1.

### Description

Under certain conditions, the FAR (Fault Address Register) and ESR (Exception Syndrome Register) could have incorrect values when the LDP (Load Pair) 64-bit variant generates an abort.

### Configurations affected

All configurations are affected.

### Conditions

All of the following conditions must be met:

1. The LDP 64-bit variant crosses a page boundary.
2. Data read from lower bytes on the first page results in a synchronous abort due to an internal ECC error, external abort, or MTE tag check fail.
3. Data read from upper bytes on the second page results in alignment fault, MMU fault, or watchpoint exception.

### Implications

If the above conditions are met the contents of the FAR and ESR registers could have incorrect values.

### Workaround

This erratum can be avoided by setting CPUACTLR5\_EL1[10] to 1. Setting CPUACTLR5\_EL1[10] to 1 would have a significant performance (approximately 20%) impact on streaming workloads that use LDP 64-bit variant. Performance impact on non streaming workloads would be much smaller (approximately 1%).

## Category A (rare)

There are no errata in this category.

## Category B

2002655

### Embedded Trace of WFI or WFE instructions might corrupt PE architectural state

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Executing a **WFI** or **WFE** instruction with Embedded Trace enabled might corrupt AArch32 PSTATE.ITSTATE, AArch64 PSTATE.BTYPE, or trace information intended for Embedded Trace, resulting in architecture violations.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

The erratum occurs under the following conditions:

1. Embedded Trace is enabled.
2. A **WFI** or **WFE** instruction is executed.

#### Implications

Under certain internal timing conditions exacerbated by a high frequency of branch instructions and/or AArch32 IT instructions, the PE might corrupt PSTATE.ITSTATE, PSTATE.BTYPE, or trace information. Corruption of any of these values might lead to any of the following architecture violations:

- Applying conditionality erroneously to AArch32 instructions (wrong condition code applied, or condition code applied when it should not be, or failing to apply a condition code when it should be).
- Applying erroneous BTYPE information to AArch64 instructions in guarded pages (wrong BTYPE value resulting in erroneous Branch Target Exceptions or failing to report a Branch Target Exception).
- Reporting erroneous trace information to Embedded Trace.

#### Workaround

This erratum 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. There is no performance or power impact associated with this workaround.

```
LDR x0,=0x6
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSEL3_EL3, X0
LDR x0,=0xF3A08002
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFFF0F7FE
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x40000001003ff
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0

LDR x0,=0x7
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSEL3_EL3, X0
LDR x0,=0xBF200000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFEF0000
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x40000001003f3
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0

ISB
```

## 2009478

### RAS errors during core power down might cause a deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

If a Reliability, Availability, and Serviceability (RAS) error occurs when a core is being powered down, then the power down sequence is designed to be aborted so that the error can be handled by software. As explained in this erratum, the power down sequence might complete despite the error occurring, or it might cause a deadlock.

#### Configurations affected

All configurations are affected.

#### Conditions

1. The ERXCTLR\_EL1.ED bit is set for any of the core error records, and at least one of the CI, DUI, CFI, FI, or UI bits is set.
2. Software running on the core sets the CPUPWRCTLR.CORE\_PWRDN\_EN bit and executes a **WFI** instruction.
3. The core Power Policy Unit (PPU) requests that the core transitions from the ON power mode to either the OFF or OFF\_EMU power mode.
4. During the L1 or L2 cache clean and invalidate done by the power transition, a RAS error occurs that causes any of the **nCOREFAULTIRQ**, **nCOREERRIRQ**, **nCOMPLEXFAULTIRQ**, or **nCOMPLEXERRIRQ** pins to be asserted.

#### Implications

The PPU will see the power down request being denied because of the RAS error, however the core will not wake up from the **WFI** instruction and therefore software is not able to handle the error. If the PPU requests the core to power down again, either because it is set to dynamic mode, or because the component programming the PPU requests the power down again, then the second power down might either:

- Complete and power off the core, despite the fact that the error has not been handled.
- Deadlock, preventing the power down from completing.

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate because of this erratum.

## Workaround

The ERXCTLR\_EL1.ED bit should be cleared for all the core error records before software sets the CPUPWRCTLR.CORE\_PWRDN\_EN bit to request a power down. This will cause any error detected during the power down to be ignored. In systems that are particularly concerned about errors during this time, software can clean and invalidate the L1 and L2 caches before clearing the ERXCTLR\_EL1.ED bit. This will minimise, but not eliminate, the probability of detecting an error during the powerdown, but at the expense of a longer time to execute the power down sequence.

## 2017090

### TRBE and SPE writes to MTE tagged pages might not report external aborts

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Memory writes due to the Trace Buffer Extension (TRBE) or Statistical Profiling Extension (SPE) to memory pages that are Tagged via the Memory Tagging Extension and present in the PE caches might fail to report external aborts.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

1. PE enables the TRBE or SPE features.
2. TRBE or SPE issues a write to a memory page that is marked MTE Tagged.
3. Interconnect returns a completion response with an error indication - Poison, DErr, or NDErr.

#### Implications

If the above conditions are met, the PE might fail to record the external abort in TRBSR\_EL1.EA (TRBE) or PMBSR\_EL1 (SPE).

#### Workaround

If the external abort conditions are non-transient, a read of the TBRE or SPE memory buffers will observe the same external abort condition as the write and take a data abort. If the external abort conditions are transient, memory pages used for TBRE and SPE can be marked as Untagged.

## 2025414

### Streaming STG and STG2 performance lower than expected with TCF=NONE

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When writing MTE allocation tags, STG and STG2 instructions might encounter lower than expected performance when SCTLR\_ELx.TCF is set to NONE.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. PE executes a series of STG or STG2 instruction with SCTLR\_ELx.TCF set to NONE.

#### Implications

If the above conditions are met, the measured bandwidth might be one half of what is expected.

#### Workaround

Disable store issue prefetching by setting CPUECLTR\_EL1[8] to 1. Note that doing so might incur a performance penalty of 0 to 0.3%.

## 2036776

### Reads of ERRIDR\_EL1.NUM return an incorrect value

#### Status

Fault Type: Programmer Category B  
Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Reads of ERRIDR\_EL1.NUM may incorrectly return a value of 0x1. The correct value is 0x2.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The PE reads ERRIDR\_EL1.

#### Implications

If the above conditions are met, the value read for ERRIDR\_EL1.NUM may incorrectly report as 0x1.

#### Workaround

Software should ignore the value in ERRIDR\_EL1.NUM, and always consider ERRIDR\_EL1.NUM to be 0x2.

## 2048466

### Utility Bus register accesses to reserved addresses of PE might hang

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Utility Bus register accesses to some reserved regions of the PE might be ignored and result in a hang.

#### Configurations Affected

All configurations are affected.

#### Conditions

This erratum occurs under the following condition:

- Utility Bus Read/Write to a reserved region in PE is executed.

#### Implications

The Utility Bus register access to DSU might not complete, resulting in a hang.

#### Workaround

Utility Bus reserved regions of page size in the PE should not be mapped by an MMU onto system memory. Also, these regions should not be made accessible by lower ELs than host.

When using an external debugger, accesses to Utility Bus reserved addresses should be avoided.

## 2067956

### Loss of MTE tag coherency with L2 cache tag ECC errors

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Dirty MTE tags contained in the L1 cache might be lost when performing streaming writes in imprecise MTE checking mode and specific timing conditions occur that coincide with an L2 tag single-bit ECC error.

#### Configurations Affected

This erratum affects all configurations of Matterhorn and the 512kB L2 configuration of Perseus when using imprecise MTE checking mode.

#### Conditions

1. PE issues streaming write to L2 cache.
2. Specific timing conditions and an L2 tag single-bit ECC error cause an L1 eviction to provide updated MTE tags from the L1 cache to the same address as the streaming write while older MTE tags are being read from the L2 cache.

#### Implications

If the above conditions are met, in rare timing conditions the updated MTE tags from the L1 cache might be lost and the streaming write might fail its MTE tag check.

#### Workaround

Set CPUACTLR\_EL1[46] to force L2 tag ECC inline correction mode. This mode incurs a 1-cycle latency penalty and thus degrades CPU performance by less than 1%.

## 2067961

### The trace data is not flushed completely during a TSB instruction executed in prohibited region

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When Embedded Trace Extension (ETE) is in trace prohibited region, and a TSB instruction is executed, any trace data associated with the instructions before the TSB must be observable in memory. Because of this erratum, four bytes of trace data are not pushed to the memory before completing the TSB instruction and those bytes might get lost or might get written to memory in next context.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. ETE is enabled.
2. ETE is in trace prohibited region.
3. TRBE is enabled.
4. TSB instruction is executed.

#### Implications

Four bytes of trace data before TSB instruction might get lost or might get written to memory in next context.

#### Workaround

To avoid this erratum, software can execute two sequential TSB instructions in any code where one TSB instruction is expected to be required.

## 2138953

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2, r0p3. Open.

#### Description

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

#### Configurations Affected

All configurations are affected.

#### Conditions

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

#### Implications

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

#### Workaround

- Workaround option 1:  
If the following code surrounds the MSR, it will prevent the erratum from happening:
  - `cpp`
  - `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:
  - `ectlr2[14:11], PF_MODE= 4'b1001`

## 2138956

### Executing a WFI or WFE instruction after a STREX instruction might result in a deadlock under specific conditions

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain micro-architectural conditions, executing a WFI or WFE instruction after a STREX that has encountered an external abort might result in a deadlock.

#### Configurations Affected

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

#### Conditions

1. Memory tagging is enabled.
2. A STREX instruction executed in MTE precise mode encounters an external abort or poisoned MTE tag.
3. A WFI or WFE instruction is executed.

#### Implications

If the above conditions are met, then, under specific micro-architectural conditions, the core might deadlock.

#### Workaround

This erratum can be avoided by inserting a sequence of 16 DMB ST instructions prior to WFI or WFE. Performance impact is expected to be negligible in real systems. This sequence can be implemented through execution of the following code at EL3 as soon as possible after boot:

```
LDR x0,=0x3
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0xF3A08002
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFF0F7FE
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x10002001003FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
LDR x0,=0x4
```

```
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0xBF200000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFEF0000
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x10002001003F3
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
```

ISB

## 2138958

### Execution of ST2G instructions in close proximity might cause loss of MTE allocation tag data

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain micro-architectural conditions, an ST2G instruction that crosses a 32-byte boundary might not write part of the MTE allocation tag to memory.

#### Configurations Affected

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

#### Conditions

1. Memory tagging is enabled.
2. Two or more ST2G instructions are executed in close proximity to the same cache line such that they miss in the L1 cache.
3. One of the ST2G instructions crosses a 32-byte boundary.

#### Implications

If the above conditions are met, then under specific micro-architectural conditions, the ST2G that crosses the 32-byte boundary might not write part of the MTE allocation tag to memory.

#### Workaround

This erratum can be avoided by setting CPUACTLR5\_EL1[13] to 1. Setting CPUACTLR5\_EL1[13] to 1 is expected to result in a small performance degradation for workloads that use MTE (approximately 1.6% when using MTE imprecise mode, 0.9% for MTE precise mode).

## 2139208

### Trace data might get overwritten in TRBE FILL mode

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Trace Buffer memory size is defined using base pointer and limit pointer in the Trace Buffer Extension (TRBE) programming model. In trace buffer fill mode, TRBE is expected to generate an interrupt and stop the collection of trace after reaching the limit pointer. Due to this erratum, under some microarchitecture conditions, TRBE might roll back to the base pointer after generating an interrupt and continue to write at the base pointer, and up to three cache lines after the base pointer before the collection stops.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. ETE and TRBE are enabled.
2. ETE is in a trace allowed region.
3. TRBLIMITR\_EL1[2:1] is programmed to 2'b00.

#### Implications

Due to this erratum, trace data present at the base pointer location and up to three cache lines after the base pointer might get overwritten. The current write pointer also increments by same number of cache line locations.

#### Workaround

Software can program 256 bytes of ignore packets starting from the base pointer and offset the write pointer TRBPTR\_EL1 by 256 bytes before enabling TBE. That ensures oldest trace is not corrupted.

## 2180384

### Enabling TRBE might cause a data write to a page with the wrong ASID when owning Exception level is EL1

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

The Trace Buffer Extension (TRBE) can be used by software to store trace packets from Embedded Trace Extension (ETE) unit to memory. The TRBE unit interfaces with the MMU for translating a virtual address to a physical address. Once a physical address is available, the TRBE unit sends trace packets to the L2 unit to be stored to the memory. The TRBE unit requests a new translation to the MMU when a virtual address crosses the 4K page boundary. Due to this erratum, if a pending translation request from Exception level EL0 or EL1 is serviced after the PE switches context to Exception level EL2, then translation with an incorrect ASID might be provided to the TRBE unit. This can lead to a write to a page with the incorrect ASID.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The TRBE is enabled.
2. Owing Exception level is EL1.
3. TRBLIMITR\_EL1.nVM is set to 0, such that the trace buffer pointer addresses are virtual addresses in the EL1&0 translation regime using the current ASID from TTBRx\_EL1. This means that the page is marked nG (non-global page).
4. The TRBE unit requests a memory translation request.
5. Before the above memory translation request completes, a context switch occurs from EL0 or EL1, to EL2.

#### Implications

If the above conditions are met, under certain microarchitectural conditions, incorrect physical address and page attributes from a different ASID might be provided to the TRBE unit. The TRBE might then write to memory using incorrect page attributes from another ASID, leading to a write that is not expected.

#### Workaround

The software should use global pages ( $nG=0$ ) for the pages that are used by the TRBE to store data when owning Exception level is EL1.

## 2189731

### The CPP instruction will apply to an incorrect EL context

#### Status

Fault Type: Programmer Category B

Fault Status: Present in rOp0. Fixed in rOp1.

#### Description

The **CPP** instruction will not operate on the desired EL as encoded in the instruction.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following condition:

1. A **CPP** instruction is executed.

#### Implications

The **CPP** instruction will cause the hardware prefetcher to invalidate the hardware prefetcher state associated with an EL other than the EL that is encoded in the instruction.

#### Workaround

Set CPUACTLR5\_EL1[44] which will cause the **CPP** instruction to invalidate the hardware prefetcher state trained from any EL.

## 2242400

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

#### Status

Fault Type: Programmer Category B  
Fault Status: Present in r0p0. Fixed in r0p1.

#### 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 setting CPUACTLR5\_EL1[17] to 1 and 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,=0x2
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0x10F600E000
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0x10FF80E000
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x80000000003FF
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
ISB
```

## 2242415

### A CFP instruction might not invalidate the correct resources

#### Status

Fault Type: Programmer Category B  
Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Executing a CFP instruction under certain conditions might not invalidate resources specified by the instruction.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. A CFP instruction is executed.
2. The Exception level specified in the instruction is EL0.
3.  $HCR\_EL2.TGE==1$  and  $HCR\_EL2.E2H==1$ .

#### Implications

If the previous conditions are met, then the CFP instruction might not invalidate branch predictor resources associated with EL0 context managed by EL2.

#### Workaround

This erratum can be avoided by setting  $CPUACTLR\_EL1[22]=1$ . Setting  $CPUACTLR\_EL1[22]$  will cause the CFP instruction to invalidate all branch predictor resources regardless of context.

Using this workaround might cause the PE to encounter another erratum. Please refer to erratum ID 2307064 "ELR\_ELx[63:48] might hold incorrect value when PE disables address translation" for more details.

## 2253138

### TRBE might cause a data write to an out-of-range address which is not reserved for TRBE

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Trace buffer memory size is defined using base pointer and limit pointer in Trace Buffer Extension (TRBE) programming model. TRBE is expected to wrap to base pointer without crossing the limit pointer. Because of this erratum, under some conditions, TRBE might generate a write to the next virtually addressed page following the last page of TRBE address space.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Embedded Trace Extension (ETE) and TRBE are enabled.
2. ETE is in trace allowed region.
3. TRBE current pointer is at last page of Trace buffer.
4. TRBE requests translation for the last page.
5. LS indicates to TRBE that it is unable to service the translation request.

#### Implications

When previous conditions are met under rare microarchitectural conditions, TRBE might incorrectly generate a data write to the next virtually addressed page following the last page of Trace Buffer. This can lead to data corruption if that page is currently used by another application and result in loss of trace up to 64 bytes.

#### Workaround

The software can mark as not valid the next page following the last TRBE page, meaning the errant access will generate a Translation Fault buffer management event. This will prevent the data corruption but will not prevent the loss of trace data.

## 2280757

### A CFP instruction might execute with incorrect upper ASID or VMID bits

#### Status

Fault Type: Programmer Category B  
Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

The upper 8 bits of ASID or VMID might be incorrect for a CFP instruction.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. A CFP is executed with ELO target execution context and {HCR\_EL2.TGE, HCR\_EL2.E2H} is {1,1} and TCR\_EL2.AS=0.
2. A CFP is executed at EL2 or EL3 and the target execution context is ELO or EL1 and VTCR\_EL2.VS=0.

#### Implications

If either of the previous conditions are met, then the CFP instruction might not invalidate branch predictor resources associated with ELO or EL1 contexts.

#### Workaround

This erratum can be avoided by setting CPUACTLR\_EL1[22]=1. Setting CPUACTLR\_EL1[22] will cause the CFP instruction to invalidate all branch predictor resources regardless of context.

## 2326639

### Denied power down request might prevent completion of future power down request

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

If a Processing Element (PE) initiates a power down request that is ultimately denied due to an external event, a future power down request might fail to complete.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. PE initiates a power down request (ON to OFF state transition) by setting CORE\_PWRDN\_EN and executing a WFI instruction.
2. PE completes the hardware flush of its caches.
3. An event, such as an external interrupt, causes an abort of the power down request.
4. PE returns to the ON state without performing a hardware reset.

#### Implications

If the above conditions are met, the PE might fail complete a subsequent power down request resulting in a deadlock.

#### Workaround

This erratum can be avoided by setting CPUACTLR2\_EL1[36] to 1 before the power-down sequence that includes setting the CORE\_PWRDN\_EN bit and executing a WFI. This bit should be cleared on exiting WFI by any mechanism other than reset.

## 2346952

### Static and dynamic TXREQ limiting might cause deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, and r0p2. Fixed in r0p3.

#### Description

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

#### Configurations Affected

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

#### Conditions

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

#### Implications

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

#### Workaround

Do not enable static or dynamic TXREQ limiting functions by keeping CPUECTLR2\_EL1[2] at 0b0 and CPUECTLR2\_EL1[1:0] at 0b00. These are the reset values.

## 2376738

### 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, r0p1, r0p2, r0p3. Open.

#### Description

A *Processing Element* (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

This erratum occurs under the following conditions:

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

#### Implications

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

#### Workaround

Set CPUACTLR2\_EL1[0] to 1 to force PLDW/PFRM ST to behave like PLD/PFRM 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.

## 2388448

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

A continuous stream of incoming DVM syncs may cause TRBE to prevent the CPU from forward progressing, while executing a WFX.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

The erratum occurs if all the following conditions are met:

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

#### Implications

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

#### Workaround

There is no workaround.

## 2388450

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r0p1.

#### 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.

## 2728475

### 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, r0p1, r0p2. Fixed in r0p3.

#### 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.

## 2743014

### 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, r0p1, and r0p2. Fixed in r0p3.

#### 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.

## 2743089

### The core might deadlock during powerdown sequence

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, and r0p2. Fixed in r0p3.

#### 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 power-down sequence as described in TRM.

#### Implications

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

#### Workaround

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

## 2779511

### 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, r0p1, r0p2. Fixed in r0p3.

#### 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.

## 3003140

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

When a DRPS instruction is executed in Debug Halt state, a double fault should cause implicit ESB according to the *Arm Architecture Reference Manual for A-profile architecture* when (SCR\_EL3.EA == '1' && SCR\_EL3.NMEA == '1' && PSTATE.EL == **EL3**). However, the *Processing Element* (PE) will only execute part of the instruction for this case.

#### Configurations affected

This erratum affects all configurations with double fault extension.

#### Conditions

This erratum occurs under the following conditions:

1. The PE is in Debug Halt state.
2. Software is currently executing at EL3 Exception level.
3. SCTLR\_EL3.IESB == '0'
4. SCR\_EL3.EA == '1' && SCR\_EL3.NMEA == '1' indicating double fault.

#### Implications

The DRPS instruction is not executed correctly.

#### Workaround

When executing a DRPS instruction in EL3, set SCTLR\_EL3.IESB to override double fault. Doing this will force the correct DRPS execution sequence to occur.

## 3022731

### TRBE 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. Fixed in r0p1.

#### Description

The *Trace Buffer Extension* (TRBE) uses the Stage-1 translation regime of the owning exception level in the owning Security state. Due to this erratum, the TRBE 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 TRBE.

#### Conditions

This erratum occurs under the following conditions:

1. TRBE is enabled.
2. TBBPTR\_EL1 and TBLIMITR\_EL1 are configured to include a virtual address VA\_X.
3. TBLIMITR\_EL1.nVM is 0.
4. A valid Stage-1 translation exists for the virtual address VA\_X.
5. If Stage-2 is enabled, a valid Stage-2 translation exists for the intermediate physical address IPA\_X for the virtual address VA\_X.
6. 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.
7. 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 TRBE might write to VA\_X rather than generating a fault. This might allow malicious software with control over TRBE 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 TRBE unless the hypervisor can handle writes to any pages mapped at Stage-2.

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

## 3031178

### 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 rOp0, rOp1, rOp2 and rOp3. 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.

## 3099213

### PE might execute instructions consistent with previous context-synchronized state when SCR\_EL3.EEL2 is changed

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2, and r0p3. Open.

#### Description

When SCR\_EL3.EEL2 is modified to a different value and a context synchronization event occurs, the PE might execute instructions consistent with previous context-synchronized state of SCR\_EL3.EEL2.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

1. The field SCR\_EL3.EEL2 is changed to a different value than last context-synchronized value.
2. A context synchronization event occurs.
3. Execution of any instruction with a behavior that depends on the value of SCR\_EL3.EEL2.

#### Implications

If the previous conditions are met, instructions might use control information saved consistent with the previous context, and might result in unexpected exceptions and load/store alignment sizes.

#### Workaround

This issue can be worked around by changing the value of any of these fields in SCR\_EL3 at the same time as changing the value of the field EEL2:

1. SCR\_EL3.EA
2. SCR\_EL3.API
3. SCR\_EL3.NMEA

Alternatively, execute the following code sequence after changing SCR\_EL3.EEL2, and prior to returning to a lower EL:

```
// Toggle the value of SCR_EL3.EA, context synchronize, then restore the value of SCR_EL3
MRS x0, SCR_EL3
LDR x1, =0x8
```

```
EOR x2, x0, x1
MSR SCR_EL3, x2
ISB
MSR SCR_EL3, x0
```

## 3324339

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in rOp0, rOp1, rOp2, and rOp3. 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 a *Speculation Barrier* (SB) instruction to ensure that the new value of PSTATE.SSBS affects subsequent instructions in the execution stream under speculation.

A kernel at EL1 or EL2 should not advertise the presence of MRS/MSR instructions to read/write the SSBS register from 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.

## 3696250

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

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

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

or

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

#### Implications

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

#### Workaround

Where software which manages the translation tables cannot ensure that it is not subject to the stall conditions, or where stalling is unacceptable, software which manages the translation tables should ignore **ID\_AA64MMFR2\_EL1.BBM** and always follow a break-before-make approach.

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

## 3701773

### Read of ICH\_VMCR\_EL2.VBPR1 might return incorrect data based on SCR\_EL3.NS

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

When ICH\_VMCR\_EL2.VBPR1 is written in Secure state (SCR\_EL3.NS==0) and then subsequently read in Non-secure state (SCR\_EL3.NS==1), a wrong value might be returned. The same issue exists in the opposite way: write in Non-secure state and read in Secure state. ICH\_VMCR\_EL2.VBPR1 is an alias of ICV\_BPR1\_EL1 which is architecturally defined as NOT banked. The RTL erroneously has this register implemented as two separate registers (secure and non-secure copies) banked by SCR\_EL3.NS.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs if all the following conditions apply:

1. The *Processing Element* (PE) is executing at EL3
2. SCR\_EL3.NS == 1 or 0
3. The PE executes an MSR ICH\_VMCR\_EL2.VBPR1 instruction
4. SCR\_EL3.NS == 0 or 1 (the opposite value from when the MSR occurred)
5. The PE executes an MRS <dst>, ICH\_VMCR\_EL2.VBPR1 instruction

#### Implications

If the previous conditions are met, the MRS <dst>, ICH\_VMCR\_EL2.VBPR1 instruction will erroneously return the value that was last written to this field with the opposite SCR\_EL3.NS value from which it was read (or the reset value if it was never written in that security state).

#### Workaround

The workaround is for EL3 software that performs context save/restore on a change of Security state to use a value of SCR\_EL3.NS when accessing ICH\_VMCR\_EL2 that reflects the Security state that owns the data being saved or restored. For example, EL3 software should set SCR\_EL3.NS to 1 when saving or restoring the value ICH\_VMCR\_EL2 for Non-secure (or Realm) state. EL3 software should clear SCR\_EL3.NS to 0 when saving or restoring the value ICH\_VMCR\_EL2 for Secure state.

## Category B (rare)

2340933

### AMBA CHI TXREQ starvation due to unfair PCrdGrant assignment

#### Status

Fault Type: Programmer Category B (rare)  
Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

Set CPUACTLR5\_EL1[61] to 1.

## 2986650

**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, r0p1, r0p2, and r0p3. 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 `VTCCR_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

1982438

### Noncompliance with prioritization of Exception Catch debug events

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r0p3. 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 EL<sub>y</sub>, where  $y > x$ , it should check the ELR\_EL<sub>y</sub> and SPSR\_EL<sub>y</sub> values to determine whether the exception was taken on an EL<sub>x</sub> exception vector address, meaning an Exception Catch on entry to EL<sub>x</sub> has been missed.

## 1982439

### 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

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r0p3. Open.

## Description

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

## Configurations Affected

All configurations are affected.

## Conditions

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

## Implications

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

## Workaround

There is no workaround.

## 1982440

### 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, r0p1, r0p2, r0p3. 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.

## 2002685

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

The erratum occurs under the following conditions:

1. Core is in Debug state.
2. SCTLR\_ELx.IESB is set to 1 for the current exception level.
3. The **DRPS** instruction is executed.

#### Implications

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

#### Workaround

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

## 2009482

### Execution of STG instructions in close proximity might incorrectly write MTE Allocation Tag to memory more than once

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain micro-architectural conditions, an STG instruction might write MTE Allocation Tag to memory more than once.

#### Configurations Affected

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

#### Conditions

1. Memory tagging is enabled.
2. Two or more STG instructions that write both Allocation Tag and Data are executed in close proximity.
3. Above STG instructions access the same cache line address but different 32 bytes in memory.

#### Implications

If the above conditions are met, then under specific micro-architectural conditions Allocation Tag for the entire cache line may be written to memory twice. This is not expected to be an issue as Allocation Tags are written by only one software agent at a time. The value of Allocation Tag will not change between these two writes since there will not be another write to Allocation Tag from another PE.

#### Workaround

There is no workaround.

## 2025410

### DSB might not guarantee completion of direct reads of L2 cache memories

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Direct reads of internal memories are implemented using a SYS #6, C15, C0, #0, <Xt> instruction executed in EL3. In response to this instruction, the PE reads an internal memory and places the contents in implementation defined DDATAx registers. A DSB is intended to guarantee completion of the memory access. If the targeted RAMs are the L2 tag, L2 victim, or L2 data RAM, a DSB might not provide this guarantee.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. PE executes a SYS #6, C15, C0, #0, <Xt> instruction, where Xt specifies the L2 tag, L2 data, or L2 victim RAMs.
2. PE executes a DSB.
3. PE reads the DDATAx registers.

#### Implications

If the above conditions are met, along with specific microarchitectural conditions, the DDATAx registers may not be updated when the DSB completes.

#### Workaround

When performing direct memory accesses to L2 cache memories do the following:

1. Set CPUACTLR2\_EL1[46].
2. Perform the direct memory accesses.
3. Clear CPUACTLR2\_EL1[46].

Note that setting CPUACTLR2\_EL1[46] incurs a 1-2% performance penalty. Thus, CPUACTLR2\_EL1[46] should not be set by default.

## 2033429

### Corrupted register state results from executing specific form of SEL instruction followed by SVE AESMC or AESIMC instruction

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain conditions, execution of two SVE instructions, either SEL(vectors) followed by AESMC or SEL(vectors) followed by AESIMC, might result in corruption of a destination vector register.

#### Configurations affected

This erratum affects configurations with CRYPTO==TRUE.

#### Conditions

The following SVE instruction sequence is required:

1. PE executes SEL <Zd>.B, <Pg>, <Zn>.B, <Zm>.B
2. PE executes either AESMC <Zd>.B, <Zd>.B or AESIMC <Zd>.B, <Zd>.B

#### Implications

The sequence of instructions described in the conditions above are not expected to occur in real code, because they do not perform useful computation. As such, no impact is expected to real systems.

#### Workaround

Because the code sequences for this erratum are not expected to occur in real code, no workaround is required.

## 2048467

### 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. Fixed in r0p1.

#### Description

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

#### Configurations Affected

The erratum affects configurations with COHERENT\_ICACHE=0.

#### Conditions

This erratum occurs under the following conditions:

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

#### Implications

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

#### Workaround

This erratum has no workaround.

## 2052733

### ER1PFGCTL.MV is improperly implemented as Read/Write

#### Status

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

#### Description

ER1PFGCTL.MV is improperly implemented as Read/Write. The Architecture requires it to be Read-As-One/Write-Ignore because all injected errors set ER1STATUS.MV=0x1.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. The PE executes a write to ER1PFGCTL.
2. The PE executes a read from ER1PFGCTL.

#### Implications

Software cannot use ER1PFGCTL.MV to determine how to control the MV bit for an injected error. All injected errors will set ER1STATUS.MV=0x1.

#### Workaround

No workaround is expected to be required.

## 2067959

### External aborts for streaming writes to MTE tagged pages may report multiple errors

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

If a streaming write to a memory page with Memory Tagging Extension (MTE) tagging enabled receives an External abort, it may report multiple SError aborts.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

1. The Processing Element (PE) generates a write of a full cache line to a memory page that is marked MTE Tagged.
2. A transaction issued on the CHI interconnect on behalf of the streaming write receives an External abort from the interconnect

#### Implications

If the above conditions are met, the PE might report multiple SError aborts. External aborts received from the interconnect represent error conditions in the system, such as accesses to unmapped devices and uncorrectable ECC errors. These conditions are more severe than the possibility of the PE reporting multiple SError aborts.

#### Workaround

There is no workaround.

## 2067960

### Trace data lost during collection stop in TRBE

#### Status

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

#### Description

When a collection stop event occurs in the Trace Buffer Extension (TRBE), trace collection from the Embedded Trace Extension (ETE) is stopped and the data in the TRBE buffers are flushed to memory. When this occurs under certain micro-architectural timing conditions, 64 bytes of trace data might get lost and replaced with Ignore bytes. This does not result in a current pointer mismatch.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. ETE and TRBE are enabled.
2. ETE is in a trace-allowed region.
3. A collection stop event occurs in TRBE.

#### Implications

When the above conditions are met, 64 bytes of trace data might get lost and replaced with Ignore bytes while getting written to memory.

#### Workaround

The erratum has no workaround.

## 2067962

### L3D\_CACHE\_ALLOC PMU inaccurate when using WriteEvictOrEvict transactions

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When the L2 cache issues a WriteEvictOrEvict transaction for a L2 copyback, the L3D\_CACHE\_ALLOC PMU event is not counted, even though it might cause an L3 cache allocation.

#### Configurations Affected

This erratum affects all configurations that enable WriteEvictOrEvict transactions through CPUECLTR\_EL1[45]=1.

#### Conditions

1. CPUECLTR\_EL1[45] is set to 1 (default value).
2. PE is configured to count PMU event 0x29 L3D\_CACHE\_ALLOCATE.
3. L2 victimizes a cache line in the UC or SC state and generates a WriteEvictOrEvict transaction.

#### Implications

If the above conditions are met, the PE fails to increment the PMU count.

#### Workaround

The DSU L3 cache PMU feature can be used to count L3 allocations across all PEs in the cluster.

## 2072986

### 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. Fixed in r0p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. A CPU executes an MSR instruction to update any of following SPR registers:
  - a. DBGBCR<n>\_EL1
  - b. DBGBVR<n>\_EL1
  - c. DBGWCR<n>\_EL1
  - d. DBGWVR<n>\_EL1
  - e. OSECCR\_EL1
2. An external debugger initiates an APB write operation for any of following registers:
  - a. DBGBCR<n>
  - b. DBGBVR<n>
  - c. DBGBXVR<n>
  - d. DBGWCR<n>
  - e. DBGWVR<n>
  - f. DBGWXXVR<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.

## 2138952

### 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. Fixed in r0p1.

#### Description

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

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

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

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

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

## 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.

## 2138954

### Incorrect Fault Status code reported for predicated SVE op

#### Status

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

#### Description

An SVE predicated store to a page with MTE tagging enabled that encounters a poisoned MTE Tag with no active elements and also has a tag check fail for a different MTE Tag might report a Synchronous External Abort instead of a Synchronous Tag Check Fault.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

This erratum occurs under the following conditions:

1. A PE executes an SVE predicated store to a page that is marked MTE Tagged.
2. The store accesses more than one Tag granule, such that there are no active elements corresponding to one of the granules accessed that also has a poisoned MTE tag.
3. The memory access results in a Tag check fail.

#### Implications

If the above conditions are met, the PE might report a Synchronous External Abort instead of Synchronous Tag Check Fault.

#### Workaround

This erratum has no workaround.

## 2138955

### Tag check fail might not be reported for an unaligned predicated SVE store

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

An SVE predicated store to a page with MTE tagging enabled might not report a fault when it encounters a tag check fail.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted

#### Conditions

This erratum occurs under the following conditions:

1. A PE executes an SVE predicated store to a page that is marked MTE Tagged.
2. The store access crosses a cache line boundary.
3. Both cache line accesses encounter a tag check fault.
4. Before the fault is reported, the first cache line is snooped out and another PE modifies the tag.
5. The tag check passes for the first cache line access when the line is fetched again.

#### Implications

If the above conditions are met, the PE might not report a Synchronous Tag Check Fault. This erratum is reported as a Programmer Category C since most of the time, a tag check fault would be correctly detected.

#### Workaround

This erratum has no workaround.

## 2138959

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

#### Status

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

#### 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 CTIDEVCTL.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.

## 2138960

### Extra A-sync packet might get written to Trace Buffer in Trace prohibited region

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

An external trace analyzer can request that an A-sync packet is injected into the trace stream, which the Embedded Trace Extension (ETE) will insert when the next P0 Element is traced. Due to this erratum, this A-sync packet might incorrectly get generated and written to trace buffer memory via TRBE under the conditions mentioned below.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

The erratum occurs under the following conditions:

1. TRBE is enabled.
2. ETE is in trace prohibited region.
3. A **TSB** instruction is executed and completed.
4. TRBE is disabled.
5. A synchronization request is received on the ATB interface.
6. TRBE is enabled.

#### Implications

If the above conditions occur, an A-sync packet might go to TRBE and when a new **TSB** instruction is executed, this packet might get written to memory. Under normal usage Arm expects that this unexpected trace will have no impact.

#### Workaround

There is no workaround.

## 2139204

### Speculative access to a recently unmapped physical address previously containing page tables might occur

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

If the memory containing page tables is unmapped or cacheable attribute is changed while there are pending hardware prefetches to that table, the read requests might illegally occur after a DSB instruction.

#### Configurations Affected

All configurations are affected.

#### Conditions

- A table walk occurs.
- The hardware prefetcher generates a cacheable request to adjacent cache lines, allocating the L2 cache.
- The physical address containing the page tables is unmapped or cacheable attribute is changed.

#### Implications

If the above conditions are met, an illegal read might occur in a short window of time after the DSB instruction. Arm does not believe this will cause incorrect execution in any practical system.

#### Workaround

No workaround is required.

## 2139205

### L1D\_CACHE\_INVALID and L2D\_CACHE\_INVALID PMU events fail to increment for SnpPreferUnique and SnpPreferUniqueFwd

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When the PE receives a SnpPreferUnique or SnpPreferUniqueFwd snoop from the interconnect, it might not correctly count the L1 data cache and L2 cache invalidations that result.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. PE receives SnpPreferUnique or SnpPreferUniqueFwd from the coherent interconnect.
2. PE invalidates the L1 data cache and L2 cache.

#### Implications

If the above conditions are met, the L1D\_CACHE\_INVALID event will fail to increment and the L2D\_CACHE\_INVALID event might fail to increment. The relative infrequency of the necessary conditions means that the L1D\_CACHE\_INVALID and L2D\_CACHE\_INVALID events are still meaningful.

#### Workaround

There is no workaround.

## 2139206

### MPAM value associated with instruction fetch might be incorrect

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r0p3. Open.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

There is no workaround.

## 2139207

### Data abort on SVE first fault load might be routed to incorrect Exception level

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain conditions, data abort on SVE first fault load might be routed to incorrect Exception level.

#### Configurations Affected

All configurations are affected.

#### Conditions

All of the following conditions must be met:

1. First active lane of SVE first fault load crosses a page boundary.
2. Translation table walk for the second page generates an external abort.
3. Memory tagging is enabled and access to bytes on the first page generates a tag check fail.
4. SCR\_EL3.EA or HCR\_EL2.TEA bits are set.

#### Implications

If the above conditions are met, data abort will not get routed to the correct Exception level. If this scenario occurred at EL0/EL1/EL2 and SCR\_EL3.EA bit is set, data abort will not get routed to EL3. Likewise if this scenario occurred at EL0/EL1 and HCR\_EL2.TEA bit is set and SCR\_EL3.EA bit is not set, data abort will not get routed to EL2. The potential impact of this erratum to a practical system is considered to be very minor, given the precondition of an unrecoverable error.

#### Workaround

There is no workaround for this erratum.

## 2149120

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

#### Status

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

#### 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.

## 2153906

### 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. Fixed in r0p1.

#### 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.

## 2189737

### PMU Event MEM\_ACCESS\_CHECKED\_WR, 0x4026 counts incorrectly and MEM\_ACC\_CHECKED 0x4024 might be incorrect

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

The PMU Event MEM\_ACCESS\_CHECKED\_WR, 0x4026 does not count correctly, and MEM\_ACC\_CHECKED 0x4024 might not count correctly.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. One of the PMU event counters is configured to count event 0x4026 or 0x4024.
2. MTE is enabled.
3. SCTLR\_ELx.ATA=1.
4. A store instruction is executed that generates a memory-write access that is Tag Checked.

#### Implications

The counter values for these events will not be correct and therefore cannot be used reliably.

#### Workaround

There is no workaround.

## 2189738

### Direct access of L2 data RAMs using RAMINDEX returns incomplete data

#### Status

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

#### Description

A direct access to the L2 data RAM using the RAMINDEX function returns incomplete data in the DDATA2 register.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following condition:

1. Direct access to internal memory targeting L2 data RAM is executed.

#### Implications

A direct access to the L2 data RAM will result in zeros on DDATA2\_EL3[19:16]. These bits should contain ECC[15:12] corresponding to Data[127:64], but instead contains zeros.

#### Workaround

There is no workaround.

## 2189739

### PMU\_HOVFS event not always exported when self-hosted trace is disabled

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

PMU\_HOVFS is a PMU event that can be exported to the ETM.

This event should be exported if self-hosted trace is disabled, or, if self-hosted trace is enabled and TRFCR\_EL2.E2TRE == 0b1.

This event is not exported if self-hosted trace is enabled and TRFCR\_EL2.E2TRE == 0b0.

Due to this erratum, when self-hosted trace is disabled, the event is never exported if TRFCR\_EL2.E2TRE == 0b0.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The ETM is configured to use PMU\_HOVFS as an external input event.
2. Self-hosted trace is disabled and TRFCR\_EL2.E2TRE == 0b0.

#### Implications

Overflows of PMU counters reserved by EL2 might not be visible.

#### Workaround

To use the PMU\_HOVFS as an external input event when self-hosted trace is disabled, ensure TRFCR\_EL2.E2TRE is set to 0b1.

## 2218242

### Some SVE PMU events count incorrectly

#### Status

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

#### Description

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

- 0x8074, SVE\_PRED\_SPEC, SVE predicated Operations speculatively executed
- 0x8075, SVE\_PRED\_EMPTY\_SPEC, SVE predicated operations with no active predicates, Operations speculatively executed
- 0x8076, SVE\_PRED\_FULL\_SPEC, SVE predicated operations with all active predicates, Operations speculatively executed
- 0x8077, SVE\_PRED\_PARTIAL\_SPEC, SVE predicated operations with partially active predicates, Operations speculatively executed

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

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

#### Implications

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

#### Workaround

There is no workaround.

## 2218243

### FAR\_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

A SVE first fault contiguous load instruction that encounters both a Tag Check fail when accessing the first active element and a watchpoint match on one of the non-first active elements can generate a Data abort exception with an incorrect value in FAR\_ELx.

#### Configurations Affected

All configurations are affected.

#### Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging and watchpoints are enabled.
2. A SVE first fault contiguous load instruction accesses memory and generates a Data Abort exception due to a Tag Check fail on the first active element.
3. There is a watchpoint match on one of the non-first active elements.

#### Implications

If the above conditions are met, a Data Abort exception will be generated with an incorrect value in FAR\_ELx. ESR\_ELx will indicate Synchronous Tag Check Fault. The FAR\_ELx value could be anything between the start address of the access and up to twice the access size.

#### Workaround

This erratum has no workaround.

## 2240288

### Incorrect sampling of SPE events "Partial predicate" and "Empty predicate" for SVE instruction with no vector operands

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain circumstances, the SPE events E[17] "Partial predicate" and E[18] "Empty predicate" might not be captured as required.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions:

1. SPE samples an SVE instruction with no vector operands.

#### Implications

If the above conditions are met, then the SPE events E[17] "Partial predicate" and E[18] "Empty predicate" might not be captured for the given instruction.

#### Workaround

There is no workaround.

## 2242397

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

#### Implications

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

#### Workaround

This erratum can be avoided by mapping all write-back memory as Inner or Outer Shareable and asserting the BROADCASTOUTER pin.

## 2242404

### 64 bit source SVE PMULLB/T not considered Cryptography instruction

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

64 bit source element variants of SVE PMULLB and PMULLT are incorrectly classified as non-cryptography instructions. When the **CRYPTODISABLE** pin is asserted, 64 bit source SVE PMULLB or SVE PMULLT instructions are executed rather than taking the expected undefined instruction exception. In addition to this, when the CRYPTODISABLE pin is deasserted, PMU counts for CRYPTO\_SPEC (PMU event 0x77) do not include 64 bit source SVE PMULLB and PMULLT in their counts.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

Cryptodisable

1. **CRYPTODISABLE** pin is high.
2. 64 bit source SVE PMULLB or SVE PMULLT is executed.

PMU Counts

1. **CRYPTODISABLE** pin is low.
2. PMU Enabled to count PMU EVENT 0x77 (CRYPTO\_SPEC).
3. 64 bit source SVE PMULLB or SVE PMULLT is executed.

#### Implications

If the above conditions are met, then the instructions will be executed instead of taking the undefined exception that is required by Arm architecture.

In addition, the PMU counter for the CRYPTO\_SPEC event (PMU EVENT 0x77) will not increment for 64 bit source SVE PMULLB PMULLT instructions.

#### Workaround

There is no workaround.

## 2242416

### 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. Fixed in r0p1.

#### 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.

## 2253154

### ESR\_ELx contents for a Data Abort exception might be incorrect when an L1D tag double bit error is encountered

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When an L1D tag double bit error is encountered, a Data Abort exception might be reported with an incorrect fault type of Synchronous Tag Check Fault in the ESR\_ELx register under unusual micro architectural conditions.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging is enabled.
2. A precise checked access due to a load instruction encounters L1D tag double bit error.

#### Implications

If the previous conditions are met, a Data Abort exception will be generated with an incorrect Data Fault Status Code (DFSC) of Synchronous Tag Check Fault in the ESR\_ELx register when it should have been Synchronous External Abort.

If this scenario occurred at EL0/EL1/EL2 and SCR\_EL3.EA bit is set, then Data Abort will not get routed to EL3.

Likewise if this scenario occurred at EL0/EL1 and HCR\_EL2.TEA bit is set, then Data Abort will not get routed to EL2. A fatal RAS error will still be reported.

#### Workaround

This erratum has no workaround.

## 2276004

### Incorrect Fault Status Code reported on a SPE buffer translation fault

#### Status

Fault Type: Programmer Category C.

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain circumstances, the 'Fault Status Code' in PMBSR\_EL1 on a buffer translation fault might be incorrectly reported as an "Address size fault, level 0 of translation or translation table base register" (FSC = 6'b00\_0000).

#### Configurations Affected

This erratum affects all configurations.

#### Conditions:

1. *Statistical Profiling Extension* (SPE) sampling and SPE buffer is enabled.
2. *Processing Element* (PE) is in Secure State.
3. SPE Owning exception level is Secure EL2.
4. *Memory Management Unit* (MMU) is enabled for EL2 or EL2&0 stage 1 address translation (SCTLR\_EL2.M = 1).
5. Buffer translation encounters a translation fault.

#### Implications

If the above conditions are met, then the fault status code in PMBSR\_EL1 is incorrectly reported as 6'b00\_0000 (Address size fault, level 0 of translation or translation table base register) instead of 6'b00\_01xx (Translation fault at level xx) on an SPE buffer translation fault.

#### Workaround

There is no workaround.

## 2280302

### Read or write from Secure EL1 for ICV\_BPR1\_EL1 register might not work

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Valid access to ICV\_BPR1\_EL1 from Secure EL1 when ICV\_CTLR\_EL1.CBPR is set to 1 should modify ICV\_BPRO\_EL1 on writes and return the value from ICV\_BPRO\_EL1 on reads. Instead, reads of ICV\_BPR1\_EL1 return ICV\_BPRO\_EL1 plus one, saturated to 0b111. Writes to ICV\_BPR1\_EL1 are ignored.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. The PE is in Secure state in the EL1 exception level.
2. ICV\_CTLR\_EL1.CBPR is set to 1.
3. A valid read or write access to ICV\_BPR1\_EL1 occurs.

#### Implications

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

#### Workaround

This erratum has no workaround.

## 2280397

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

#### Status

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

#### 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.

## 2307062

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### 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.

## 2307063

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

#### Status

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

#### 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.

## 2307064

### ELR\_ELx[63:48] might hold incorrect value when PE disables address translation

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When the CPU executes an exception return in order to switch context and the new context satisfies certain rare conditions, the top 16 bits of ELR\_ELx might track an incorrect value.

#### Configurations Affected

The erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. CPUACTLR\_EL1[22] is set to 1.
2. The PE executes an ERET, ERETAA or ERETAB instruction to switch to a new context.
3. Either stage 1 or stage 2 translation was enabled when ERET is executed. After ERET, both stage 1 and stage 2 translations are turned off.
4. ELR\_ELx[63:48] specified by ERET is neither 0x0000 (all zero) nor 0xffff (all one).

#### Implications

When the above conditions are met, the PE takes instruction abort (address size fault) or asynchronous exception after ERET without executing the instruction in the context specified by ERET. After the exception is taken, ELR\_ELx specified by ERET should hold the same value because no instruction is executed. However, PE might modify ELR\_ELx[63:48] to zero.

ERET with non-zero ELR\_ELx[63:48] causes an address size fault during address translation disabled because the CPU supports less than 256TB physical address space. Arm also assumes the new context is controlled by privileged software (for example, Hypervisor) because translation is turned off. Therefore, software can hit this erratum only when the system software uses this malicious address in the ELR\_ELx register.

#### Workaround

This erratum has no workaround.

## 2307065

### TRBE might use incorrect Cacheability attributes for TRBE data when address translation is disabled

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

Under certain conditions, Trace Buffer Extension (TRBE) might use incorrect Cacheability attributes for TRBE data when address translation is disabled.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. TRBE is enabled with `TRBLIMITR_EL1.nVM == 1`.
2. `TRBMAR_EL1.Attr` is programmed to use Cacheable attributes.
3. `MDCR_EL2.E2TB = 2'b00` (EL2 owning)
4. `HCR_EL2.CD=1` and `HCR_EL2.VM=1`
5. PE is executing at EL=1 or EL=0.
6. TRBE writes data to memory.

#### Implications

When the above conditions are met, PE might incorrectly use Non-Cacheable attribute instead of Cacheable attribute from `TRBMAR_EL1.Attr[3:0]` for TRBE data. Trace data might be lost if the memory location used by TRBE is present in cache when this write happens.

#### Workaround

This erratum has no workaround.

## 2307066

### L1 MTE Tag poison is not cleared

#### Status

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

#### Description

The MTE Tag poison is not cleared by an STG or DC G[Z]VA instruction.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

This erratum occurs under the following conditions:

1. A Processing Element (PE) accesses a line that encounters poison on the MTE Tag.
2. The PE executes an STG or DC G[Z]VA to the same 16-byte address.

#### Implications

If the above conditions are met, then the MTE Tag poison does not get cleared in the L1 Tag.

#### Workaround

There is no workaround.

## 2341660

### ESR\_ELx contents for a Data Abort exception might be incorrect when a data double bit error or external abort is encountered

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When a data double bit error or external abort is encountered on one half of an unaligned load, a Data Abort exception might be reported with an incorrect fault type of Synchronous Tag Check Fault in the ESR\_ELx register. This occurs under unusual micro-architectural conditions.

#### Configurations Affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging is enabled.
2. A precise checked access due to an unaligned load instruction encounters a data double bit error or external abort.

#### Implications

If the previous conditions are met, a Data Abort exception will be generated with an incorrect Data Fault Status Code (DFSC) of Synchronous Tag Check Fault in the ESR\_ELx register, when it should have been Synchronous External Abort.

If this scenario occurred at EL0/EL1/EL2, and the SCR\_EL3.EA bit is set, then the Data Abort will not get routed to EL3.

Likewise, if this scenario occurred at EL0/EL1, and the HCR\_EL2.TEA bit is set, then the Data Abort will not get routed to EL2. A RAS error will still be reported.

#### Workaround

This erratum has no workaround.

## 2341661

### 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. Fixed in r0p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under all of the following conditions:

1. Data double bit error or external abort is encountered during translation table walk and synchronous exception is reported.

#### Implications

If the previous conditions are met, ESR\_ELx.ISV bit will be set.

#### Workaround

This erratum has no workaround.

## 2341669

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

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

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

or

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

#### Implications

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

- SPE event E[17] "Partial predicate"
- SPE event E[18] "Empty predicate"
- PMU event SVE\_PRED\_EMPTY\_SPEC (0x8075)
- PMU event SVE\_PRED\_FULL\_SPEC (0x8076)
- PMU event SVE\_PRED\_NOT\_FULL\_SPEC (0x8079)
- PMU event SVE\_PRED\_PARTIAL\_SPEC (0x8077)

## Workaround

This erratum has no workaround.

## 2341671

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

#### Description

When a permission fault or unsupported atomic fault is detected in the second stage of translation during stage 1 translation table walk, and there is a higher priority alignment fault due to SCTLR\_EL1.C bit not being set, then Data Abort might be generated reflecting the lower priority fault.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs when all the following conditions apply:

1. The core executes an atomic, load/store exclusive, or load-acquire/store-release instruction.
2. SCTLR\_EL1.C bit is not set and access is not aligned to size of data element.
3. A permission fault or unsupported atomic fault is detected in the second stage of translation during stage 1 translation table walk.

#### Implications

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

#### Workaround

This erratum has no workaround.

**2346744****CSSELR\_EL1.TnD is RAZ/WI when CSSELR\_EL1.InD == 0x1****Status**

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

**Description**

In some contexts when CSSELR\_EL1.InD == 0x1, CSSELR\_EL1.TnD is defined to be RES0.

In other contexts when CSSELR\_EL1.InD == 0x0, CSSELR\_EL1.TnD is defined to be R/W.

When a bit is RES0 in some contexts and R/W in other contexts, then it cannot be implemented as RAZ/WI for RES0 contexts.

In affected products, CSSELR\_EL1.TnD is incorrectly treated as RAZ/WI instead of the correct R/W behavior when CSSELR\_EL1.InD == 0x1.

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

This erratum occurs under the following conditions:

1. The PE is executing with CSSELR\_EL1.InD == 0x1.
2. The PE attempts to read or write CSSELR\_EL1.TnD.

**Implications**

Reads of CSSELR\_EL1.TnD will return 0x0 and writes will be ignored.

**Workaround**

This erratum is not expected to require a workaround.

**2346745**

## **L2 tag RAM double-bit ECC error might lead to the PE not responding to a forwarding snoop**

### **Status**

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r0p1.

### **Description**

A double-bit ECC error in a cache line containing Memory Tagging Extensions (MTE) tags might result in the L1 and L2 caches becoming out-of-sync with respect to MTE tag validity. This can lead to a situation in which the L1 evicts dirty MTE tags to the L2 as part of a fill/evict sequence or a snoop. If this eviction satisfies an external forwarding snoop, the RN-F might fail to provide legal responses which might lead to a deadlock.

### **Configurations Affected**

This erratum affects all configurations using the Memory Tagging Extensions.

### **Conditions**

When using MTE, under specific microarchitectural and timing conditions, an L2 double-bit ECC error in the L2 tag RAMs might allow the L1 data cache to later evict a cache line with dirty MTE tags.

The erratum occurs if the eviction satisfies an external snoop of one of these types:

- SnpUniqueFwd
- SnpCleanFwd
- SnpSharedFwd
- SnpNotSharedDirtyFwd
- SnpPreferUniqueFwd

### **Implications**

If the previous conditions are met, the PE might provide an SnpRespDataFwded response to the HN-F, but fail to provide a CompData response to the original requester, leading to a system deadlock.

### **Workaround**

This erratum has no workaround.

## 2393116

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

#### Status

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

#### 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.

## 2446315

### 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, r0p1, r0p2, r0p3. 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.

## 2446526

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, and r0p1. Fixed in r0p2.

#### 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.

**2639704****STALL\_BACKEND\_MEM, Memory stall cycles AMU event count incorrectly****Status**

Fault Type: Programmer Category C  
Fault Status: Present in r0p3. Open.

**Description**

The following *Activity Monitor Unit* (AMU) event does not count correctly:

- 0x4005, STALL\_BACKEND\_MEM. The counter counts cycles in which the PE is unable to dispatch instructions from the frontend to the backend of the PE. It is due to a backend stall caused by a miss in the last level of cache within the PE clock domain. This event is counted by AMEVCNTR03.

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

- AMU is enabled

**Implications**

The counter values for the event will not be correct and therefore cannot be used reliably.

**Workaround**

This erratum has no workaround.

## 2648408

### ERXPFGCDN\_EL1 register is incorrectly written on Warm reset

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, and r0p2. Fixed in r0p3.

#### Description

The ERXPFGCDN\_EL1 register is written a reset value of 0 at both cold and Warm reset, when it should only be reset at Cold reset.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs when a Warm reset occurs.

#### Implications

If the previous condition is met, the value of ERXPFGCDN\_EL1 will not be preserved across a Warm reset.

#### Workaround

This erratum has no workaround.

## 2648410

### Incorrect read value for Performance Monitors Control Register

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, and r0p2. Fixed in r0p3.

#### 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.

## 2664787

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2. Fixed in r0p3.

#### 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.

## 2675383

### FAR\_ELx contents for a Data Abort exception on SVE first fault contiguous load instruction due to Tag Check fail might be incorrect

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2. Fixed in r0p3.

#### Description

A *Scalable Vector Extension* (SVE) first fault contiguous load instruction that encounters a Tag Check fail when accessing the first active element and a watchpoint match on one of the non-first active elements can generate a Data abort exception with incorrect value in FAR\_ELx.

#### Configurations Affected

All configurations are affected.

#### Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging and watchpoints are enabled.
2. An SVE first fault contiguous load instruction accesses memory and generates a Data Abort exception due to Tag Check fail on the first active element.
3. There is a watchpoint match on one of the non-first active elements.

#### Implications

If the above conditions are met, a Data Abort exception will be generated with an incorrect value in FAR\_ELx. ESR\_ELx will indicate Synchronous Tag Check Fault.

#### Workaround

This erratum has no workaround.

## 2696764

### Execution of STG instructions in close proximity might cause loss of MTE allocation tag data

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2. Fixed in r0p3.

#### Description

Under certain rare micro-architectural conditions, two or more STG instructions that access the same cacheline but different 32-bytes might not write the *Memory Tagging Extension* (MTE) allocation tag to memory in the presence of an ECC error to the same cache index.

#### Configurations Affected

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

#### Conditions

This erratum occurs under the following conditions:

1. Memory tagging is enabled.
2. Two or more STG instructions are executed in close proximity to the same cache line.
3. The STG instructions access different 32-bytes locations.
4. An L2 fill for a different cacheline but to the same index has a single bit data error that could have otherwise caused a capacity evict of the cacheline accessed by the STG instructions.

#### Implications

If the above conditions are met, then under specific micro-architectural conditions, the MTE allocation tag might not be written to memory, resulting in a silent corruption of the MTE tag.

#### Workaround

If desired, this erratum can be avoided by setting CPUACTLR5\_EL1[13] to 1.

Note: setting CPUACTLR5\_EL1[13] to 1 is expected to result in a small performance degradation for workloads that use MTE (approximately 1.6% when using MTE imprecise mode, 0.9% for MTE precise mode).

## 2728476

### MTE checked load might read an old value of allocation tag by not complying with address dependency ordering

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r0p3 Open.

#### Description

Under some unusual micro-architectural conditions, checked load might read an old value of allocation tag by not complying with address dependency ordering.

#### Configurations Affected

All configurations are affected.

#### Conditions

The erratum occurs when all the following apply:

1. Initially, memory location M has allocation tag A.
2. Processing Element x (PE<sub>x</sub>) stores to M using allocation tag A.
3. PE<sub>y</sub> changes the allocation tag of M from A to B.
4. PE<sub>x</sub> makes a checked load from M using allocation tag A, with a dependency such that it should observe allocation tag B.

#### Implications

If the above conditions are met, PE<sub>x</sub> may not observe the new allocation tag for the memory location and may fail to report a tag check fail.

#### Workaround

This erratum has no workaround.

## 2728477

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2. Fixed in r0p3

#### 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.

**2736656**

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

### Status

Fault Type: Programmer Category C.

Fault Status: Present in r0p0, r0p1, r0p2, r0p3. Open.

### Description

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

### Configurations Affected

This erratum affects all configurations.

### Conditions

This erratum occurs under the following conditions:

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

### Implications

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

### Workaround

There is no workaround.

## 2738454

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, and r0p2. Fixed in r0p3.

#### Description

The following *Performance Monitoring Unit* (PMU) events are over-counted by the number of cycles in which the PMU counters are enabled:

- 0x3E, STALL\_SLOT\_FRONTEND, no operation sent for execution on a slot due to the frontend
- 0x3F, STALL\_SLOT no operation sent for execution on a slot

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

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

- 0x3E, STALL\_SLOT\_FRONTEND
- 0x3F, STALL\_SLOT

#### Implications

When a PMU counter is configured to count the 0x3E event (STALL\_SLOT\_FRONTEND) or the 0x3F event (STALL\_SLOT), the value counted will be greater than expected, with the overage equal to the number of cycles in which the PMU counters are enabled.

#### Workaround

To work around this erratum, software that utilizes the 0x3E (STALL\_SLOT\_FRONTEND) or 0x3F (STALL\_SLOT) count can correct the value that was read by subtracting from it a count of cycles in which the PMU counters were enabled, which is event 0x11 (CPU\_CYCLES).

## 2755354

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, and r0p2. Fixed in r0p3.

#### 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.

## 2794918

### DGH instruction doesn't execute correctly

#### Status

Fault Type; Programmer Category C

Fault Status: Present in rOp0, rOp1, rOp2 and rOp3. Open

#### Description

DGH instructions are executed as PSBs. The DGH target address is ignored.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. A DGH instruction is executed.

#### Implications

If Profiling is not enabled, then the PSB will execute as a NOP. Performant code sequences that depend on DGH for explicit memory management will not see the expected speedup, but will see no additional slowdown.

If Profiling is enabled, then the PSB instruction might take up to tens of cycles to complete, causing an additional slowdown.

Since neither DGH nor PSB affect the architected state, there is no functional problem.

#### Workaround

No workaround is expected to be necessary.

## 2803644

### Incorrect decoding of SVE version of PRF\* scalar plus scalar instructions

#### Status

Fault Type: Programmer Category C

Fault Status: Present in rOp0, rOp1, rOp2, rOp3. Open.

#### Description

Scalar plus Scalar forms of the *Scalable Vector Extension* (SVE) PRF may not prefetch from the correct address. The address should be  $X_n + X_m \ll \text{scalar}$ , but is instead calculated as  $X_n$ . This affects the following instructions:

- PRFB (scalar plus scalar)
- PRFH (scalar plus scalar)
- PRFW (scalar plus scalar)
- PRFD (scalar plus scalar)

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Any of the above instructions are executed without trapping when  $X_m \neq 0x0$

#### Implications

All affected instructions are software prefetches which do not affect architectural state in any way (including suppression of any translation faults). Thus this erratum will not affect the functional operation of the CPU. Since these instructions are likely to be used in contexts where  $X_n$  is fixed and  $X_m$  is incrementing, it is unlikely that the erroneous prefetches would result in undesired cache pollution or reduction in memory bandwidth because the instructions will simply continuously prefetch the same address.

#### Workaround

No workaround is expected to be necessary, but if one is specifically needed, the programmer can use an ADD, and then one of the immediate forms of SVE PRF, which are unaffected. These instructions are:

- PRFB (scalar plus immediate)
- PRFH (scalar plus immediate)

- PRFW (scalar plus immediate)
- PRFD (scalar plus immediate)

## 2814366

### ECC errors in MTE allocation tags may lead to silent data corruption in tag values

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, and r0p3. Open.

#### Description

Streaming writes that require *Memory Tagging Extension* (MTE) tags for tag checking or merging with data receive allocations tags that are flagged as poisoned may lead to the *Processing Element* (PE) caching data and tags with no indication that the tags are poisoned. This may lead to silent data corruption on the allocation tags.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. The PE performs a streaming write (a write of 64 contiguous bytes gathered from multiple store or DC ZVA operations).
2. Streaming write requires MTE tag check or hits in the PE caches to a line that contains MTE allocation tags.
3. MTE allocations tags contain an indication of an error (uncorrectable ECC error or poison flag).

#### Implications

If the above conditions are met, the PE might merge the streaming write data and the MTE allocation tags containing an error and write data and allocation tags to a cache without marking the tags as poisoned. This can lead to silent data corruption to future consumers of the MTE allocation tags, which may result in incorrect MTE tag check results. The net effect is an increase in the SDC FIT rate of the PE.

There is still substantial benefit being gained from the ECC logic.

#### Workaround

There is no workaround.

## 2814415

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r0p3. Open.

#### Description

The timestamp value that is captured in the *Statistical Profiling Extension* (SPE) records may be incorrect.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. Timestamp capture is enabled for SPE records at the appropriate Exception level 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 stale (off by one tick) or zero in some cases.

#### Workaround

There is no workaround.

## 2814419

### 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, r0p1, r0p2, and r0p3. 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.

## 2820250

### TRBE buffer write translation out of context may have incorrect memory attributes

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, and r0p3. Open.

#### Description

When `TRBLIMITR_EL1.nVM = 1`, `TBE_OWNING_EL = EL1`, and TRBE requests a translation while the *Processing Element* (PE) is executing in EL2 or EL3, and cache is disabled by `HCR_EL2.CD = 1`, memory attribute may not be Non-cacheable.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. `TRBLIMITR_EL1.nVM` is set to 1.
2. `MDCR_EL2.E2TB` is set to 0b10 or 0b11.
3. `HCR_EL2.CD` is set to 1.
4. The PE is executing in EL2 or EL3.
5. TRBE requests a translation for a buffer write.

#### Implications

Memory attributes for any write access by TRBE to that translation may not be forced to Non-cacheable.

#### Workaround

Use of `HCR_EL2.CD` is not expected to be common. If a workaround is needed, do not allow TRBE to be given to a VM machine.

## 2914113

### Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

A PE accessing a same physical memory location with mismatched Shareability attributes and requiring a read of *Memory Tagging Extension* (MTE) tags might result in data corruption.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. PE accesses a physical memory location using cacheable and Non-shareable attributes.
2. PE accesses the same physical address using cacheable and shareable attributes with MTE checking enabled.

#### Implications

If the previous conditions are met, the PE might expose stale data from the PE caches established by a Non-shareable access. This data might become visible to shareable observers in the same Shareability domain, even if the PE performs the required cache maintenance for ensuring ordering and coherency when aliasing Shareability.

#### Workaround

Arm expects that operating systems do not use mismatched Shareability attributes for aliases of the same memory location for tagged pages.

## 2933583

### L2D\_CACHE\_WB\_CLEAN overcounts

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

Counting of the L2D\_CACHE\_WB\_CLEAN event includes transfer of data directly to another *Processing Element* (PE) using the AMBA CHI Direct Cache Transfer mechanism.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

1. The PE processes a forwarding snoop from the DSU or Fully coherent Home Node (HN-F) and sends data directly to another PE using a CompData message.

#### Implications

If the previous condition is met, the PE will count the L2D\_CACHE\_WB\_CLEAN event contrary to the architectural specification of this event.

#### Workaround

No workaround is required for this erratum.

## 2973112

### IRG instructions might produce the wrong tag when GCR\_EL1.RRND=0x0.

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

When the *Processing Element* (PE) is configured with GCR\_EL1.RRND=0x0, writing SCTLR\_EL3.ATA, SCTLR\_EL2.ATA, SCTLR\_EL1.ATA, or SCTLR\_EL1.ATA0 can corrupt internal state. As a result IRG instructions might produce the wrong tag.

#### Configurations Affected

This erratum affects all configurations with MTEDISABLE=0x0.

#### Conditions

This erratum occurs under the following conditions:

1. The PE is executing with GCR\_EL1.RRND=0x0.
2. An IRG instruction is executed.
3. An MSR is executed which updates any of SCTLR\_EL3.ATA, SCTLR\_EL2.ATA, SCTLR\_EL1.ATA, or SCTLR\_EL1.ATA0.
4. An IRG instruction is executed.

#### Implications

If the above conditions are met, the tag produced by the second or any subsequent IRG instruction might be incorrect.

#### Workaround

Arm is not aware of any software which uses the GCR\_EL1.RRND=0x0 configuration. If your system uses this configuration, please contact Arm Customer Support for more information.

## 3070040

### TagMatch responses with error indication do not generate a SError abort

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, and r0p3. Open.

#### Description

When tag checks are performed outside of the *Processing Element* (PE), the AMBA CHI protocol returns a TagMatch response that indicates whether or not the tag check succeeded or failed. If an error condition occurred while performing the tag check, the system might return the TagMatch response with an error indication. If this occurs, the PE should report a SError abort, but fails to do so.

#### Configurations affected

This erratum affects all configurations with the BROADCASTMTE pin asserted.

#### Conditions

This erratum occurs under the following conditions:

1. PE has *Memory Tagging Extension* (MTE) enabled in asynchronous checking of stores.
2. PE performs tag checked stores.
3. Write streaming causes the PE to send the stores to the interconnect as write transactions.
4. While performing the tag check operation for the write, the interconnect encounters an error condition while reading the tag value.

#### Implications

If the conditions are met, the interconnect might return a TagMatch response with an error indication, but the PE might not generate a SError abort. If the TagMatch response indicates a tag check failure (Resp=Fail), TFSR\_ELx bits will still be updated.

#### Workaround

No workaround is required for this erratum.

**3604862****PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative****Status**

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

**Description**

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

**Configurations affected**

This erratum affects all configurations.

**Conditions**

The erratum occurs if the following condition applies:

- MSR PSTATE.{PAN or UAO} executes

**Implications**

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

**Workaround**

A workaround is not expected to be required.

## 3605043

### Incorrect count for PMU event 0x004C (L1D\_TLB\_REFILL\_RD) might be observed

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

A hardware generated prefetch operation or a PRFM instruction might indicate a L1D\_TLB\_REFILL\_RD event leading to an incorrect count.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

The erratum occurs if all the following conditions apply:

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

#### Implications

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

#### Workaround

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

- Event 0x0005 (L1D\_TLB\_REFILL)
- Event 0x004D (L1D\_TLB\_REFILL\_WR)
- Event 0x10E. (L1D\_TLB\_REFILL\_RD\_PF)

These events can be used to calculate an Effective event 0x004C as follows:

Effective Event 0x004C = Event 0x0005 - Event 0x004D - Event 0x010E

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

Arm Architecture Reference Manual relevant events:

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

Implementation Defined relevant event:

Mnemonic	Number
L1D_TLB_REFILL_RD_PF	0x010E

Arm Architecture Reference Manual relevant metric:

"Attributable Level 1 TLB refill rate, read" (Event 0x004C / Event 0x004E)

## 3627241

### PMU event STALL\_SLOT\_FRONTEND counts when instruction fetch is stalled for PCRFB availability

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

When instructions are not available to be dispatched due to Program Counter Register File (PCRF) fullness, they are counted by the STALL\_SLOT\_FRONTEND PMU event instead of the STALL\_SLOT\_BACKEND PMU event.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs whenever instruction fetch is stalled due to PCRFB fullness and the PMU is configured to count the STALL\_SLOT\_FRONTEND or STALL\_SLOT\_BACKEND events.

#### Implications

Correlation of STALL\_FRONTEND and STALL\_SLOT\_FRONTEND telemetry might be impacted when the PCRFB is often full, because the STALL\_FRONTEND PMU event will not count under the same PCRFB full conditions.

#### Workaround

This erratum has no workaround.

## 3633461

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

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

#### Implications

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

#### Workaround

There is no workaround.

## 3640937

### SPE operation type is corrupted under certain conditions

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

This erratum occurs under the following conditions:

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

#### Implications

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

#### Workaround

There is no workaround.

## 3694436

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

The erratum occurs if all the following conditions apply:

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

#### Implications

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

#### Workaround

No workaround is expected to be necessary for this erratum.

## 3694458

### FFR might not capture the lowest faulting memory element

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

The erratum occurs if all of the following conditions apply:

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

#### Implications

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

#### Workaround

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

## 3700127

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

#### Configurations affected

This erratum affects all configurations.

#### Conditions

The erratum occurs if all the following conditions apply:

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

#### Implications

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

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

#### Workaround

No workaround is necessary for this erratum.

## 3705908

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r0p3. Open.

#### Description

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

PMU events are mis-categorized by failing to consider "Taken locally", specifically resulting in mis-categorizations between PMU events EXC\_UNDEF and EXC\_TRAP\_OTHER, as well as between PMU events EXC\_SVC and EXC\_TRAP\_OTHER.

#### Configurations affected

This erratum affects all configurations.

#### Conditions

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

1. When the effective value of HCR\_EL2.{E2H,TGE} **is** {1,1}, an exception can increment PMU event 0x008D EXC\_TRAP\_OTHER, when the exception should instead increment PMU event 0x0081 EXC\_UNDEF.
2. When the effective value of HCR\_EL2.{E2H,TGE} is **NOT** {1,1}, an exception can increment PMU event 0x0081 EXC\_UNDEF, when the exception should instead increment PMU event 0x008D EXC\_TRAP\_OTHER.
3. When the effective value of HCR\_EL2.{E2H,TGE} is **NOT** {1,1}, executing an SVC instruction can increment PMU event 0x0082 EXC\_SVC, when that SVC instruction should instead increment PMU event 0x008D EXC\_TRAP\_OTHER.

#### Implications

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

#### Workaround

There is no workaround.

# Proprietary notice

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 affiliates) in the US and/or elsewhere. Please follow Arm's trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(PRE-1121-V1.0)

# Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

## Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

### Product completeness status

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

### Product revision status

The rxpy identifier indicates the revision status of the product described in this manual, where:

**rx**

Identifies the major revision of the product.

**py**

Identifies the minor revision or modification status of the product.