Arm Neoverse V2 (MP158)

Software Developer Errata Notice

Date of issue: March 10, 2025

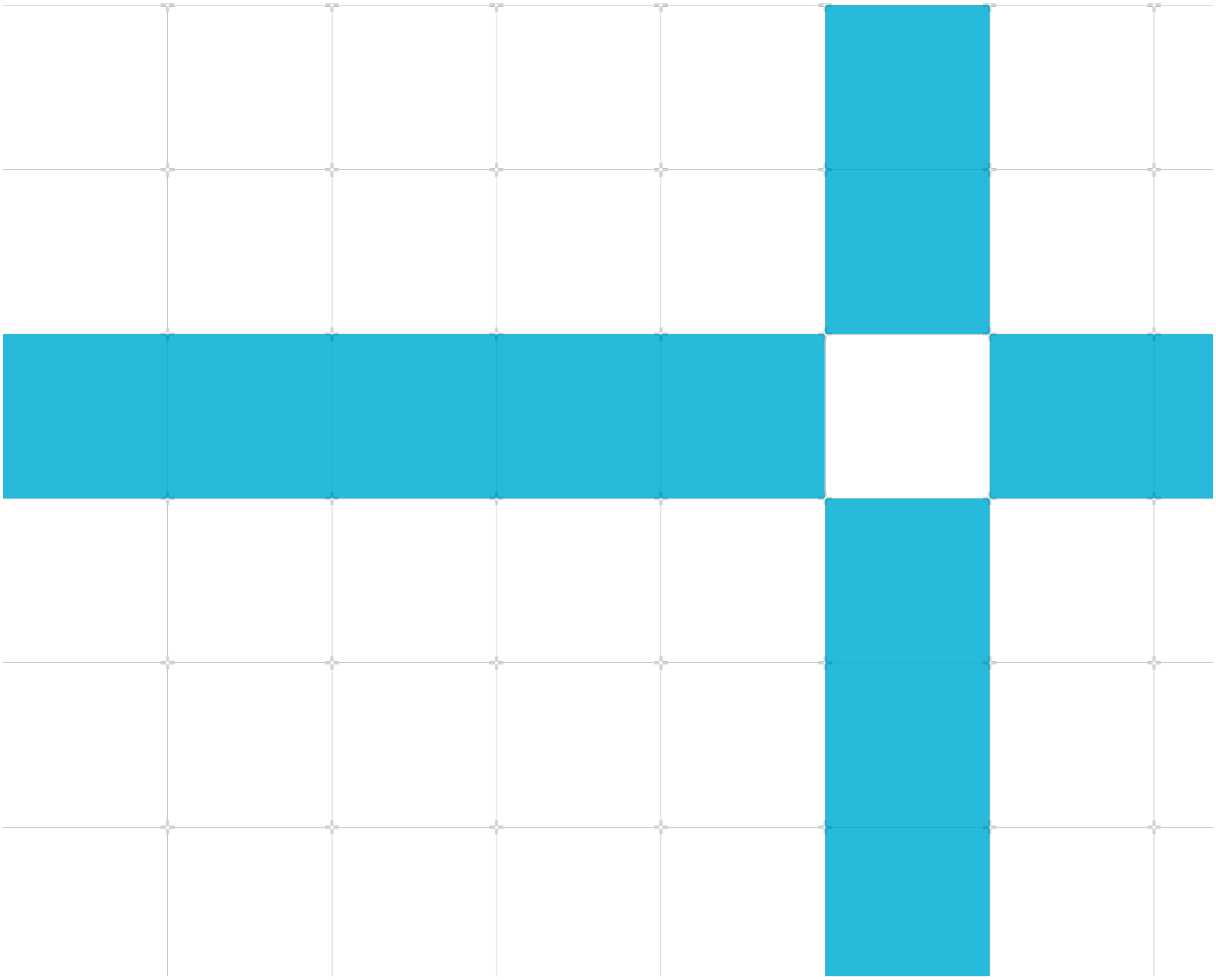
Non-Confidential

Document version: 11.0

Copyright © 2025 Arm® Limited (or its affiliates). All rights reserved.

Document ID: SDEN-2332927

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



This document is Non-Confidential.

Copyright © 2025 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_2332927_11.0_en) was issued on March 10, 2025.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-2332927**

Inclusive language commitment

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

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Neoverse V2 (MP158), create a ticket on **https://support.developer.arm.com**.

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

Contents

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.
Non-Confidential

r0p0 implementation fixes

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

REVIDR_EL1[0]	2644884 L1 hardware prefetcher might cause deadlock

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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

r0p1 implementation fixes

Note the following errata might be fixed in some implementations of r0p1. 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]	2662553 Static and dynamic TXREQ limiting might cause deadlock
REVIDR_EL1[4]	2855383 Precise abort can lead to a deadlock

Note that there is no change to the MIDR_EL1 which remains at r0p1. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Introduction

Scope

This document describes errata categorized by level of severity. Each description includes:

- The current status of the erratum.
- Where the implementation deviates from the specification and the conditions required for erroneous behavior to occur.
- The implications of the erratum with respect to typical applications.
- The application and limitations of a workaround where possible.

Categorization of errata

Errata are split into three levels of severity and further qualified as common or rare:

Category A	A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.
Category A (Rare)	A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category B	A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.
Category B (Rare)	A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category C	A minor error.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

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.

March 10, 2025: Changes in document version v11.0

ID	Status	Area	Category	Summary
3438991	New	Programmer	Category B	When Hardware Page Aggregation (HPA) is enabled memory accesses may be translated incorrectly
3696445	New	Programmer	Category B	Unprivileged data memory-dependent prefetches might leak privileged data
3841324	New	Programmer	Category B	PE might branch to an incorrect BR/BLR target

October 01, 2024: Changes in document version v10.0

ID	Status	Area	Category	Summary
3696242	New	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock
3701771	New	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS
3604860	New	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative
3605041	New	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed
3627356	New	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability
3633459	New	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception
3640931	New	Programmer	Category C	SPE operation type is corrupted under certain conditions
3694432	New	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled
3694456	New	Programmer	Category C	FFR might not capture the lowest faulting memory element
3700125	New	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors
3705906	New	Programmer	Category C	PMU events are mis-categorized by not considering the effect of "Taken locally"

April 30, 2024: Changes in document version v9.0

ID	Status	Area	Category	Summary
3324336	New	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing
3442699	New	Programmer	Category B	PE may execute incorrect instructions

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

December 15, 2023: Changes in document version v8.0

ID	Status	Area	Category	Summary
3099206	New	Programmer	Category B	PE might execute instructions consistent with previous context-synchronized state when SCR_EL3.EEL2 is changed
3070048	New	Programmer	Category C	TagMatch responses with error indication do not generate a SError abort

August 30, 2023: Changes in document version v7.0

ID	Status	Area	Category	Summary
3002998	New	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly
3031173	New	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2
2986655	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
2794917	New	Programmer	Category C	DGH instruction doesn't execute correctly
2914111	New	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption
2933584	New	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts
2985980	New	Programmer	Category C	SPE latency counters are corrupted under certain conditions
2989895	New	Programmer	Category C	IRG instructions might produce the wrong tag when GCR_EL1.RRND=0x0

March 07, 2023: Changes in document version v6.0

No new or updated errata in this document version.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

December 16, 2022: Changes in document version v5.0

ID	Status	Area	Category	Summary
2743011	New	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock
2779510	New	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation
2801372	New	Programmer	Category B	The core might deadlock during powerdown sequence
2764406	New	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP
2769032	New	Programmer	Category C	STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly
2801065	New	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions
2802338	New	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFE state
2813403	New	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM
2813408	New	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled
2814365	New	Programmer	Category C	ECC errors in MTE allocation tags may lead to silent data corruption in tag values
2817024	New	Programmer	Category C	TRBE buffer write translation out of context may have incorrect memory attributes

August 10, 2022: Changes in document version v4.0

ID	Status	Area	Category	Summary
2662553	New	Programmer	Category B	Static and dynamic TXREQ limiting might cause deadlock
2719103	New	Programmer	Category B	Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back
2719105	New	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch
2675381	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
2694799	New	Programmer	Category C	MTE tag check fail seen on first half of a cache-line crossing load does not get reported
2696811	New	Programmer	Category C	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data
2719108	New	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field
2719109	New	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED
2719111	New	Programmer	Category C	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering

May 02, 2022: Changes in document version v3.0

ID	Status	Area	Category	Summary
2394277	Updated	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption
2395412	Updated	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing
2618597	New	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag and BTB RAMs
2644884	New	Programmer	Category B	L1 hardware prefetcher might cause deadlock
2299866	New	Programmer	Category C	Incorrect read value for Performance Monitors Control Register
2446309	New	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR
2446525	New	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly
2626876	New	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register
2630907	New	Programmer	Category C	Read to dump the instruction cache contents while in Debug state results in deadlock
2640782	New	Programmer	Category C	PMU MEM_ACCESS_CHECKED_RD and MEM_ACCESS_CHECKED_WR inaccurate
2644885	New	Programmer	Category C	ERXPFGCDN_EL1 register is incorrectly written on Warm reset
2644899	New	Programmer	Category C	Incorrect sampling of SPE events "tlb_access" for an unaligned SVE load instruction with no active elements

December 17, 2021: Changes in document version v2.0

ID	Status	Area	Category	Summary
2394277	New	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption
2395412	New	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing

October 29, 2021: Changes in document version v1.0

ID	Status	Area	Category	Summary
2331132	New	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock
2331130	New	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect
2331134	New	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

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
2331132	Programmer	Category B	Disabling of data prefetcher with outstanding prefetch TLB miss might cause a deadlock	r0p0, r0p1, r0p2	Open
2394277	Programmer	Category B	Translation table walk folding into an L1 prefetch might cause data corruption	r0p0	r0p1
2395412	Programmer	Category B	A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing	r0p0	r0p1
2618597	Programmer	Category B	Entry into the Full Retention power mode might cause corruption on Itag and BTB RAMs	r0p0, r0p1	r0p2
2644884	Programmer	Category B	L1 hardware prefetcher might cause deadlock	r0p0	r0p1
2662553	Programmer	Category B	Static and dynamic TXREQ limiting might cause deadlock	r0p0, r0p1	r0p2
2719103	Programmer	Category B	Core might fetch stale instruction from memory when both Stage 1 Translation and Instruction Cache are Disabled with Stage 2 forced Write-Back	r0p0, r0p1	r0p2
2719105	Programmer	Category B	Continuous failing STREX because of another PE executing prefetch for store behind consistently mispredicted branch	r0p0, r0p1	r0p2
2743011	Programmer	Category B	Page crossing access that generates an MMU fault on the second page could result in a livelock	r0p0, r0p1	r0p2
2779510	Programmer	Category B	The PE might generate memory accesses using invalidated mappings after completion of a DVM SYNC operation	r0p0, r0p1	r0p2
2801372	Programmer	Category B	The core might deadlock during powerdown sequence	r0p0, r0p1	r0p2
3002998	Programmer	Category B	PE executing DRPS during Debug Halt under Double Fault condition will not execute properly	r0p0, r0p1, r0p2	Open

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

ID	Area	Category	Summary	Found in versions	Fixed in version
3031173	Programmer	Category B	SPE might write to pages which lack write permission at Stage-1 or Stage-2	r0p0, r0p1, r0p2	Open
3099206	Programmer	Category B	PE might execute instructions consistent with previous context-synchronized state when SCR_EL3.EEL2 is changed	r0p0, r0p1, r0p2	Open
3324336	Programmer	Category B	MSR PSTATE.SSBS to 0 is not fully self-synchronizing	r0p0, r0p1, r0p2	Open
3438991	Programmer	Category B	When Hardware Page Aggregation (HPA) is enabled memory accesses may be translated incorrectly	r0p0, r0p1, r0p2	Open
3442699	Programmer	Category B	PE may execute incorrect instructions	r0p0, r0p1, r0p2	Open
3696242	Programmer	Category B	Changing block size without break-before-make or mis-programming contiguous hint bit can lead to a livelock	r0p0, r0p1, r0p2	Open
3696445	Programmer	Category B	Unprivileged data memory-dependent prefetches might leak privileged data	r0p0, r0p1, r0p2	Open
3701771	Programmer	Category B	Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS	r0p0, r0p1, r0p2	Open
3841324	Programmer	Category B	PE might branch to an incorrect BR/BLR target	r0p0, r0p1	r0p2
2986655	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	Open
2299866	Programmer	Category C	Incorrect read value for Performance Monitors Control Register	r0p0	r0p1
2331130	Programmer	Category C	MPAM value associated with instruction fetch might be incorrect	r0p0, r0p1, r0p2	Open
2331134	Programmer	Category C	Noncompliance with prioritization of Exception Catch debug events	r0p0, r0p1, r0p2	Open
2446309	Programmer	Category C	Software-step not done after exit from Debug state with an illegal value in DSPSR	r0p0, r0p1, r0p2	Open
2446525	Programmer	Category C	PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND events count incorrectly	r0p0	r0p1

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

ID	Area	Category	Summary	Found in versions	Fixed in version
2626876	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register	r0p0	r0p1
2630907	Programmer	Category C	Read to dump the instruction cache contents while in Debug state results in deadlock	r0p0, r0p1	r0p2
2640782	Programmer	Category C	PMU MEM_ACCESS_CHECKED_RD and MEM_ACCESS_CHECKED_WR inaccurate	r0p0	r0p1
2644885	Programmer	Category C	ERXPFGCDN_EL1 register is incorrectly written on Warm reset	r0p0	r0p1
2644899	Programmer	Category C	Incorrect sampling of SPE events "tlb_access" for an unaligned SVE load instruction with no active elements	r0p0	r0p1
2675381	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
2694799	Programmer	Category C	MTE tag check fail seen on first half of a cache-line crossing load does not get reported	r0p0, r0p1	r0p2
2696811	Programmer	Category C	Execution of STG instructions in close proximity might cause loss of MTE allocation tag data	r0p0, r0p1	r0p2
2719108	Programmer	Category C	Incorrect read value for Performance Monitors Configuration Register EX field	r0p0, r0p1	r0p2
2719109	Programmer	Category C	Incorrect value reported for SPE PMU event SAMPLE_FEED	r0p0, r0p1	r0p2
2719111	Programmer	Category C	MTE checked load might read an old value of allocation tag by not complying with address dependency ordering	r0p0, r0p1	r0p2
2764406	Programmer	Category C	Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP	r0p0, r0p1	r0p2
2769032	Programmer	Category C	STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly	r0p0	r0p1
2794917	Programmer	Category C	DGH instruction doesn't execute correctly	r0p0, r0p1, r0p2	Open
2801065	Programmer	Category C	Incorrect decoding of SVE version of PRF* scalar plus scalar instructions	r0p0, r0p1	r0p2

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

ID	Area	Category	Summary	Found in versions	Fixed in version
2802338	Programmer	Category C	AMU Event 0x0011, Core frequency cycles might increment incorrectly when the core is in WFE state	r0p0, r0p1, r0p2	Open
2813403	Programmer	Category C	PE might fail to detect multiple uncorrectable ECC errors in the L1 data cache tag RAM	r0p0, r0p1	r0p2
2813408	Programmer	Category C	Incorrect timestamp value reported in SPE records when timestamp capture is enabled	r0p0, r0p1	r0p2
2814365	Programmer	Category C	ECC errors in MTE allocation tags may lead to silent data corruption in tag values	r0p0, r0p1, r0p2	Open
2817024	Programmer	Category C	TRBE buffer write translation out of context may have incorrect memory attributes	r0p0, r0p1, r0p2	Open
2914111	Programmer	Category C	Accessing a memory location using mismatched Shareability attributes when MTE tag checking is enabled might cause data corruption	r0p0, r0p1, r0p2	Open
2933584	Programmer	Category C	L2D_CACHE_WB_CLEAN overcounts	r0p0, r0p1, r0p2	Open
2985980	Programmer	Category C	SPE latency counters are corrupted under certain conditions	r0p0, r0p1, r0p2	Open
2989895	Programmer	Category C	IRG instructions might produce the wrong tag when GCR_EL1.RRND=0x0	r0p0, r0p1, r0p2	Open
3070048	Programmer	Category C	TagMatch responses with error indication do not generate a SError abort	r0p0, r0p1, r0p2	Open
3604860	Programmer	Category C	PSTATE.{PAN,UAO} synchronization might not be honored while MSR PSTATE is speculative	r0p0, r0p1, r0p2	Open
3605041	Programmer	Category C	Incorrect count for PMU event 0x004C (L1D_TLB_REFILL_RD) might be observed	r0p0, r0p1, r0p2	Open
3627356	Programmer	Category C	PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability	r0p0, r0p1, r0p2	Open
3633459	Programmer	Category C	EDSCR.STATUS not updated on Halting Step when a Load-Exclusive instruction generates a synchronous exception	r0p0, r0p1, r0p2	Open

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

ID	Area	Category	Summary	Found in versions	Fixed in version
3640931	Programmer	Category C	SPE operation type is corrupted under certain conditions	r0p0, r0p1, r0p2	Open
3694432	Programmer	Category C	LS misses RAR hazard on case with clean critical beat and poisoned final response with ECC disabled	r0p0, r0p1, r0p2	Open
3694456	Programmer	Category C	FFR might not capture the lowest faulting memory element	r0p0, r0p1, r0p2	Open
3700125	Programmer	Category C	PE might fail to log a RAS error for L2 data RAM ECC errors	r0p0, r0p1, r0p2	Open
3705906	Programmer	Category C	PMU events are mis-categorized by not considering the effect of "Taken locally"	r0p0, r0p1, r0p2	Open

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Category B

2331132
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. 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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

value indicated:

- ECTLR2[14:11], PF_MODE= 4'b1001

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2394277
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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2395412
A continuous stream of incoming DVM syncs may cause TRBE to prevent the core from forward progressing

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions are met:

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

Implications

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

Workaround

There is no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2618597
Entry into the Full Retention power mode might cause corruption on Itag and BTB RAMs

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

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

Configurations Affected

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

Conditions

The erratum occurs if all the following conditions apply:

- The *Processing Element* (PE) enters the FULL_RET power state.
- The Itag or BTB RAMs are placed into a low-power mode during the PE FULL_RET power state.
- The PE power state transitions back to ON without going through the OFF power state.

Implications

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

- Fetch and execute incorrect opcodes as a result of itag corruption.
- Predict incorrect targets from corrupted BTB RAMs.

Workaround

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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2644884
L1 hardware prefetcher might cause deadlock

Status

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

Description

Clock gating logic in the L2 cache might cause internal interface signals to remain asserted, leading to unexpected operation of one of the L1 data cache hardware prefetchers.

Configurations Affected

This erratum affects all configurations

Conditions

Hardware prefetching is enabled.

Implications

If the previous condition is met, unexpected operation, including deadlock, might occur.

Workaround

Disable the affected L1 data cache prefetcher by setting CPUACTLR6_EL1[41] to 'b1. Doing so will incur a performance penalty of ~1%.

Contact Arm for an alternate workaround that impacts power.

2662553
Static and dynamic TXREQ limiting might cause deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2719103
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 and r0p1. Fixed in r0p2.

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 makes a 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 requests 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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

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.

2719105
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 and r0p1. Fixed in r0p2.

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/PRFM LD and not cause invalidations to other PE caches. There might be a small performance degradation to this workaround for certain workloads that share data.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2743011
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 and r0p1. Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2779510
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. Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2801372
The core might deadlock during powerdown sequence

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, and r0p1. Fixed in r0p2.

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

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

Implications

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

Workaround

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

3002998
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 and r0p2. 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:

The PE is in Debug Halt state.
Software is currently executing at EL3 Exception level.
SCTLR_EL3.IESB == '0'
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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3031173
SPE might write to pages which lack write permission at Stage-1 or Stage-2

Status

Fault Type: Programmer Category B
Fault Status: r0p0, r0p1 and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3099206
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, and r0p2. 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_EL
    MRS x0,SCR_EL3
    LDR x1,=0x8
```

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

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

3324336
MSR PSTATE.SSBS to 0 is not fully self-synchronizing

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, and r0p2. 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 EL0. Arm expects that kernels provide system calls for EL0 software to modify PSTATE.SSBS when the SSBS register is not implemented and that EL0 software will use this when the presence of the SSBS register is not advertised.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3438991 When Hardware Page Aggregation (HPA) is enabled memory accesses may be translated incorrectly

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Open.

Description

When Hardware Page Aggregation (HPA) is enabled memory accesses may be translated incorrectly. This may permit bypass of Stage-2 translation.

This issue has been assigned CVE ID CVE-2024-5660.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. Hardware page aggregation is enabled (CPUECTLR_EL1[46]==0, which is the default value).
2. Stage-1 and/or Stage-2 translation is enabled for the active translation regime.
3. At Stage-1 or Stage-2 any of the following occur:
    - Translation table entries are modified to change the table or block size without following a Break-Before-Make approach.
    - Translation table entries within the same contiguous region have inconsistent values for the contiguous bit.

4. The translation table entries in condition 3 have inconsistent values for output addresses, access permissions, and/or memory attributes.

5. Complex, but not rare, microarchitectural conditions occur.

Implications

When all of the conditions above are met, any memory access translated by the translation table entries in condition 3 might use a Physical Address Space (PAS), Physical Address (PA), access permissions, and/or memory attributes which are not consistent with the architectural combination of Stage-1 translation and Stage-2 translation. Specifically any of the following may occur:

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

- The resulting PAS may be any arbitrary PAS reachable from the security state the access originated from:
    - For accesses originating from Non-secure state: Non-secure PAS only.
    - For accesses originating from Secure state: Secure or Non-secure PAS only.

- The resulting PA can be any arbitrary PA.
- The resulting access permissions can be any arbitrary access permissions.
- The resulting memory attributes can be any arbitrary memory attributes.

The resulting translation may permit software to read or write to an arbitrary PA which should not be accessible due to Stage-2 translation and/or may permit resulting memory attributes which should not be possible due to Stage-2 translation. Consequently this may allow software within a virtual machine to escalate privilege to EL2.

The resulting translation does not permit software in Normal state to read or write to any PA in the Secure PAS and consequently this does not provide a mechanism for software in Normal state to escalate privilege to Secure state.

Workaround

The erratum can be avoided by setting CPUECTLR_EL1[46] to 1, which will disable hardware page aggregation.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3442699
PE might execute incorrect instructions

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, and r0p2. Open.

Description

The PE might execute incorrect instructions when icache is enabled.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following condition applies:

- Icache is enabled

Implications

If the previous conditions are met, incorrect instructions might be executed.

Workaround

This erratum can be worked around by setting CPUACTLR_EL1[36] before enabling icache.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3696242
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 and r0p2. 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.

3696445
Unprivileged data memory-dependent prefetches might leak privileged data

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Open.

Description

An unprivileged context can trigger a data memory-dependent prefetch engine to fetch the contents of a privileged location for which it does not have read permission, and consume those contents as an address that is also dereferenced.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

- Data memory-dependent prefetch engine is trained on an adversary's data and then deployed to access a privileged location.

Implications

An unprivileged (EL0) attacker can force the prefetcher to load content from a privileged location in the same translation regime and dereference it despite the permission checks and TCR.E0PDx. This secret-dependent access might leave a trace in data caches and TLBs that could be measured by the attacker to recover the secret.

This issue does not affect guest-to-guest and guest-to-hypervisor isolation guarantees. Likewise, in configurations with RME enabled, *Granule Protection Checks* (GPC) are honoured by the prefetcher.

Workaround

The erratum can be avoided by disabling the affected prefetcher setting CPUACTLR6_EL1[41].

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3701771
Read of ICH_VMCR_EL2.VBPR1 might return incorrect data based on SCR_EL3.NS

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. 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_EL.NS value from which it was read (or the reset value if it was never written in that security state).

Workaround

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

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.

3841324
PE might branch to an incorrect BR/BLR target

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Under certain complex microarchitectural conditions, the *Processing Element* (PE) might branch to an incorrect BR (Branch to register)/BLR (Branch with link to register) target.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum might occur when all the following conditions apply:

- Memory Management Unit (MMU) is enabled.
- Execution of BR/BLR instructions.
- Complex microarchitectural conditions that can only be generated by executing context.

Implications

When all the previous conditions are met, the PE might branch to an incorrect indirect target in the same context for the BR/BLR that is executed. This incorrect indirect target would have been established by either the same polymorphic BR/BLR or other polymorphic indirect branch in the same context. This would lead to incorrect instructions within the same context being executed resulting in unpredictable behavior.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[1] prior to enabling MMU. This bit will disable a branch predictor power savings feature. Disabling this power feature results in negligible power movement and no performance impact.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Category B (rare)

2986655
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, and r0p2. Open.

Description

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

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

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

- If MDCR_EL2.TDE == 0, a stage 2 Data Abort might result in a Data Abort exception taken erroneously to EL1.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

    - The rarity of PE internal timings required to exhibit this bug is comparable to *Reliability, Availability, and Serviceability* (RAS) error FIT rates. Expected outcome is a kernel panic that will kill the process.
- If MDCR_EL2.TDE == 1, a stage 1 Data Abort might result in a Data Abort exception taken erroneously to EL2.
    - This scenario is containable within a hypervisor via the software workaround outlined below.

Workaround

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

EL2 can protect against this case as follows:

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

Category C

2299866
Incorrect read value for Performance Monitors Control Register

Status

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

Description

The Performance Monitors Control Register (PMCR_EL0) and the External Performance Monitor Control Register (PMCR) might return an incorrect read value for the X field.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Software writes a nonzero value to the PMCR_EL0.X, or debugger writes a nonzero value to the PMCR.X
2. Software reads the PMCR_EL0 register, or debugger reads the PMCR register

Implications

The PMCR_EL1.X or PMCR.X field incorrectly reports the value 0x1, indicating exporting of events in an IMPLEMENTATION DEFINED PMU event export bus is enabled. The expected value is 0x0, as the implementation does not include a PMU event export bus.

Workaround

This erratum has no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2331130
MPAM value associated with instruction fetch might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2331134
Noncompliance with prioritization of Exception Catch debug events

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2. Open.

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

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

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

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

2446309
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. Open.

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum has no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2446525
PMU STALL_SLOT_BACKEND and STALL_SLOT_FRONTEND 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:

- 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2626876
Incorrect read value for Performance Monitors Configuration Register

Status

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

Description

The Performance Monitors Configuration Register (PMCFGR) returns an incorrect read value for the CCD field.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Debugger reads the PMCFGR register.

Implications

The PMCFGR.CCD field incorrectly reports the value 0x1 indicating that Cycle counter has prescale, instead of the expected value of 0x0, since the field is RAZ if AArch32 isn't supported.

Workaround

There is no workaround.

2630907
Read to dump the instruction cache contents while in Debug state results in deadlock

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, and r0p1. Fixed in r0p2.

Description

In Debug state, an access to read the instruction cache data contents using SYS_IMP_RAMINDEX will not complete and will deadlock any ITR transactions that follow.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs if all the following conditions apply:

1. The PE enters Debug state.
2. User sets SYS_IMP_RAMINDEX RAM_ID field to 0x1 in order to select the read of instruction cache contents, and performs the read.

Implications

The instruction cache read deadlocks, and the debugger might lose control.

Workaround

This erratum can be avoided by the debugger if the instruction cache is not read when the core is in Debug state.

2640782
PMU MEM_ACCESS_CHECKED_RD and MEM_ACCESS_CHECKED_WR inaccurate

Status

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

Description

The MEM_ACCESS_CHECKED_RD and MEM_ACCESS_CHECKED_WR PMU events increment incorrectly when accessing a tagged page, but is inactive due to SVE predication.

Configurations Affected

This erratum affects configurations with BROADCASTMTE=1.

Conditions

This erratum occurs if the following conditions apply:

1. a load or store access crosses a page-boundary
2. one unaligned half accesses a page that is MTE tagged, but is inactive due to SVE predication
3. the other unaligned half accesses a page that is not MTE tagged

Implications

If the previous conditions are met, the PMU event might increment inaccurately.

Workaround

This erratum has no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2644885
ERXPFGCDN_EL1 register is incorrectly written on Warm reset

Status

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

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.

2644899
Incorrect sampling of SPE events "tlb_access" for an unaligned SVE load instruction with no active elements

Status

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

Description

Under certain circumstances, the SPE events E[4] "TLB Access" might not be captured as required.

Configurations Affected

This erratum affects all configurations.

Conditions

SPE samples an unaligned SVE load instruction with no active elements.

Implications

If the previous conditions are met, then the SPE events E[4] "TLB Access" might not be consistent with the PMU event 0x0025 (L1D_TLB). Note that PMU event 0x0025 (L1D_TLB) is accurate.

Workaround

This erratum has no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2675381
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 and r0p1. Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2694799
MTE tag check fail seen on first half of a cache-line crossing load does not get reported

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Under some unusual microarchitectural conditions, tag check fail seen on first half of a cache-line crossing load does not get reported.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under all of the following conditions:

1. Memory tagging is enabled
2. Cache-line crossing load is executed that fails tag check on first half of the access
3. Unusual microarchitectural conditions occur

Implications

If the above conditions are met, precise checked loads that see tag mismatch will not report an exception and imprecise checked loads will not update the TFSR register.

Workaround

This erratum has no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2696811
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 and r0p1. Fixed in r0p2.

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

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).

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2719108
Incorrect read value for Performance Monitors Configuration Register EX field

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2719109
Incorrect value reported for SPE PMU event SAMPLE_FEED

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1, Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2719111
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 and r0p1. Fixed in r0p2.

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 (PEx) stores to M using allocation tag A.
3. PEy changes the allocation tag of M from A to B.
4. PEx 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, PEx 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2764406
Incorrect value reported for SPE PMU event 0x4000 SAMPLE_POP

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, and r0p1. Fixed in r0p2.

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

If a workaround is desired, then minimization of potential overcounting of the SAMPLE_POP event can be realized via software disable of any PMU SAMPLE_POP event counters whenever SPE is disabled, and also upon the servicing of a PMBIRQ interrupt. For profiling of EL0 workloads, software can further reduce exposure to overcounting by configuring the counter to not count at Exception levels of EL1 or higher.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2769032
STALL_BACKEND_MEM, Memory stall cycles AMU event count incorrectly

Status

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

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2794917 DGH instruction doesn't execute correctly

Status

Fault Type; Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. 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.

2801065
Incorrect decoding of SVE version of PRF* scalar plus scalar instructions

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

Scalar plus Scalar forms of the *Scalable Vector Extension* (SVE) PRF may not prefetch from the correct address. The address should be Xn + Xm << scalar, but is instead calculated as Xn. This affects the following instructions:

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. Any of the above instructions are executed without trapping when Xm != 0x0

Implications

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

Workaround

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

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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2802338
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. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2813403
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, and r0p1. Fixed in r0p2.

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2813408
Incorrect timestamp value reported in SPE records when timestamp capture is enabled

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2814365
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, and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

2817024
TRBE buffer write translation out of context may have incorrect memory attributes

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2914111 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 and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2933584
L2D_CACHE_WB_CLEAN overcounts

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Open.

Description

Counting of the L2D_CACHE_WB_CLEAN event includes transfer of data directly to another *Processing Element* (PE) using the AMBA CHI Direct Cache Transfer mechanism.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

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

Implications

If the previous condition is met, the PE will count the L2D_CACHE_WB_CLEAN event contrary to the architectural specification of this event.

Workaround

No workaround is required for this erratum.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2985980
SPE latency counters are corrupted under certain conditions

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Open.

Description

Under certain conditions, the dispatch to issue and dispatch to completion latency counters for certain Statistical Profiling samples might be corrupted.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Statistical profiling is enabled at the appropriate Exception level.
2. The first instruction sampled is one of the following instructions:
    - FADDA
    - BFMMLA
    - FDIV
    - FSQRT
3. The sample gets flushed under certain micro-architectural conditions.
4. The next sample of one of the above instructions might capture incorrect latency values.

Implications

If the above conditions are met, the dispatch to issue and dispatch to completion counts for certain samples of FADDA, BFMMLA, FDIV, or FSQRT in the *Statistical Profiling Extension* (SPE) buffer might be corrupted.

Workaround

There is no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

2989895
IRG instructions might produce the wrong tag when GCR_EL1.RRND=0x0.

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3070048
TagMatch responses with error indication do not generate a SError abort

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

3604860
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 and r0p2. Open.

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following condition applies:

- MSR PSTATE.{PAN or UAO} executes

Implications

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

Workaround

A workaround is not expected to be required.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3605041
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 and r0p2. 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)

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3627356
PMU event STALL_SLOT_FRONTEND counts when instruction fetch is stalled for PCRF availability

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Open.

Description

When instructions are not available to be dispatched due to Program Counter Register File (PCRF) fullness, they are counted by the STALL_SLOT_FRONTEND PMU event instead of the STALL_SLOT_BACKEND PMU event.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs whenever instruction fetch is stalled due to PCRF fullness and the PMU is configured to count the STALL_SLOT_FRONTEND or STALL_SLOT_BACKEND events.

Implications

Correlation of STALL_FRONTEND and STALL_SLOT_FRONTEND telemetry might be impacted when the PCRF is often full, because the STALL_FRONTEND PMU event will not count under the same PCRF full conditions.

Workaround

This erratum has no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3633459 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 and r0p2. Open.

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

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

Implications

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

Workaround

There is no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3640931
SPE operation type is corrupted under certain conditions

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3694432
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 and r0p2. Open.

Description

When PE is configured with ERR0CTLR.ED = 0, a load instruction that received data on the CPU AMBA CHI interface with some words marked Poisoned can violate internal visibility requirement.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. PE is configured with ERR0CTLR.ED = 0, disabling Error detection and correction
2. Data requested by a load instruction is received on the CPU AMBA CHI interface with some words marked Poisoned, indicating an uncorrected error has been detected in the system
3. Load consumes non-poisoned words from the returned data.
4. Another PE performs a write to one or more of the bytes consumed by the load

Implications

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

Workaround

No workaround is expected to be necessary for this erratum.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3694456
FFR might not capture the lowest faulting memory element

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3700125
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 and r0p2. Open.

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if all the following conditions apply:

1. Error correction is enabled with ERR0CTLR.ED set to 1.
2. PE is performing simultaneous memory reads to both Device or Normal Non-cacheable and Normal-WriteBack memory.
3. Specific timing conditions occur.
4. PE detects an ECC error in the L2 data RAM.

Implications

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

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

Workaround

No workaround is necessary for this erratum.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

3705906
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 and r0p2. 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:

- 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.
- 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.
- When the effective value of HCR_EL2.{E2H,TGE} is **NOT** {1,1}, executing an SVC instruction can increment PMU event 0x0082 EXC_SVC, when that SVC instruction should instead increment PMU event 0x008D EXC_TRAP_OTHER.

Implications

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

Workaround

There is no workaround.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential

Proprietary notice

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

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

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

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and 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.

Copyright © 2025 Arm® Limited or its affiliates. All rights reserved.

Non-Confidential