arm

Arm Cortex-A77 MP074

Software Developer Errata Notice

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

Non-Confidential Proprietary notice

This document is protected by copyright and other related rights and the practice 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. **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 implementations infringe any third party patents.

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, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

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.

This document consists solely of commercial items. You shall be responsible for ensuring that any 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.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. 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 the Agreement shall prevail.

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

Copyright © 2020 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Web address

http://www.arm.com/.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:
- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on this document

If you have comments on content then send an e-mail to errata@arm.com giving:
- The document title.
- The document number: SDEN-1152370.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Contents

r0p0 implementation fixes

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

REVIDR_EL1[0]	1220737 Streaming store under specific conditions might cause deadlock or data corruption

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

r1p0 implementation fixes

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

REVIDR_EL1[0]	1316063 Modification of the translation table for a virtual page which is being accessed by an active process might lead to read after write ordering violation.

Note that there is no change to the MIDR_EL1 which remains at r1p0 but the REVIDR_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

r1p1 implementation fixes

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

REVIDR_EL1[3]	1450698 Software Step might prevent interrupt recognition

Note that there is no change to the MIDR_EL1 which remains at r1p1 but the REVIDR_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

Introduction

Scope

This document describes errata categorized by level of severity. Each description includes:
- The current status of the erratum.
- Where the implementation deviates from the specification and the conditions required for erroneous behavior to occur.
- The implications of the erratum with respect to typical applications.
- The application and limitations of a workaround where possible.

Categorization of errata

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

Category A　　　　A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.

Category A (Rare)　　　　A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.

Category B　　　　A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.

Category B (Rare)　　　　A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.

Category C　　　　A minor error.

Change control

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The errata summary table on page 15 identifies errata that have been fixed in each product revision.

01-Sep-2020: Changes in document version 11.0				
ID	Status	Area	Cat	Summary of erratum
1508412	Updated	Programmer	CatB	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
1923201	New	Programmer	CatB	External debugger access to debug registers might not work during Warm reset
1925769	New	Programmer	CatB	Store operation that encounters multiple hits in the TLB can access regions of memory with attributes that could not be accessed at that exception level or security state
1857204	New	Programmer	CatC	A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data
1869877	New	Programmer	CatC	ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect
1880115	New	Programmer	CatC	Noncompliance with prioritization of Exception Catch debug events
1884878	New	Programmer	CatC	The core might report incorrect fetch address to FAR_ELn when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified
1899210	New	Programmer	CatC	Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO
1899434	New	Programmer	CatC	PFG duplicate reported faults through a warm reset
1923198	New	Programmer	CatC	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB

19-May-2020: Changes in document version 10.0				
ID	Status	Area	Cat	Summary of erratum
1508412	Updated	Programmer	CatB	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
1800714	New	Programmer	CatB	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB

1662411	New	Programmer	CatC	Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock
1702492	New	Programmer	CatC	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB
1788065	New	Programmer	CatC	Possible loss of CTI event
1788067	New	Programmer	CatC	Loss of CTI events during warm reset
1827134	New	Programmer	CatC	External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior
1830646	New	Programmer	CatC	Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR

08-Nov-2019: Changes in document version 9.0				
ID	Status	Area	Cat	Summary of erratum
1467687	Updated	Programmer	CatB	Branch prediction for an ERET cached in the instruction cache might cause a deadlock
1508412	Updated	Programmer	CatB	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock
1542418	New	Programmer	CatB (rare)	The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches
1514033	New	Programmer	CatC	Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE
1519163	New	Programmer	CatC	AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle
1522097	New	Programmer	CatC	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error
1523503	New	Programmer	CatC	CPUECTLR_EL1 controls for the MMU have no affect
1610369	New	Programmer	CatC	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect
1624431	New	Programmer	CatC	CPUAMEVTYPER4_EL0 register cannot be written

15-Jul-2019: Changes in document version 8.0				
ID	Status	Area	Cat	Summary of erratum
1508412	New	Programmer	CatB	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock

1515815	New	Programmer	CatB	The core might execute multiple instructions before taking a software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache
1487187	New	Programmer	CatC	Waypoints from previous session might cause single-shot comparator match when trace enabled
1488613	New	Programmer	CatC	An unaligned load might initiate a prefetch request which crosses a page boundary
1491015	New	Programmer	CatC	TRCIDR3.CCITMIN value is incorrect

24-May-2019: Changes in document version 7.0

ID	Status	Area	Cat	Summary of erratum
1450698	New	Programmer	CatB	Software Step might prevent interrupt recognition
1467687	New	Programmer	CatB	Branch prediction for an ERET cached in the instruction cache might cause a deadlock

29-Mar-2019: Changes in document version 6.0

ID	Status	Area	Cat	Summary of erratum
1418842	New	Programmer	CatB (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data
1328683	New	Programmer	CatC	Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError
1346768	New	Programmer	CatC	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0
1355135	New	Programmer	CatC	L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events
1395535	New	Programmer	CatC	Read from PMCCNTR in AArch32 might return corrupted data
1405548	New	Programmer	CatC	MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit
1415321	New	Programmer	CatC	LDREX-STREX might succeed incorrectly when an intervening store occurs and LDREX detects a single-bit ECC error on the cache line in the L1 data cache tag RAM
1421023	New	Programmer	CatC	Portions of the branch target address recorded in ETM trace information are incorrect for an indirect branch with a malformed branch target address

15-Mar-2019: Changes in document version 5.0

ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

03-Dec-2018: Changes in document version 4.0

ID	Status	Area	Cat	Summary of erratum
1316063	New	Programmer	CatA (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-write ordering violation
1286809	New	Programmer	CatB (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation

05-Oct-2018: Changes in document version 3.0

ID	Status	Area	Cat	Summary of erratum
1160841	Updated	Programmer	CatB	Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock
1177367	Updated	Programmer	CatB	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation Status
1191167	Updated	Programmer	CatB	MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result
1204882	Updated	Programmer	CatB	The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence
1220737	New	Programmer	CatB	Streaming store under specific conditions might cause deadlock or data corruption
1253791	New	Programmer	CatB	Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption Status
1262841	New	Programmer	CatB	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation
1273521	New	Programmer	CatB	A T32 instruction inside an IT block followed by mispredicted speculative instruction stream might cause a deadlock
1148171	Updated	Programmer	CatC	ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array
1151664	Updated	Programmer	CatC	Direct access to internal memory for L2 TLB might not update IDATAn_EL3 registers

1162083	Updated	Programmer	CatC	16-bit T32 instruction close to breakpoint location may cause early breakpoint exception
1185469	Updated	Programmer	CatC	Exception packet for return stack match might return incorrect [E1:E0] field
1192280	Updated	Programmer	CatC	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level
1207839	Updated	Programmer	CatC	Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error
1220404	New	Programmer	CatC	Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry
1220843	New	Programmer	CatC	ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported
1244986	New	Programmer	CatC	Illegal return event might corrupt PSTATE.UA0
1256789	New	Programmer	CatC	Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error
1262908	New	Programmer	CatC	Write-Back load after two Device-nG* stores to the same physical address might get invalid data

27-Jul-2018: Changes in document version 2.0				
ID	Status	Area	Cat	Summary of erratum
1160841	New	Programmer	CatB	Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock
1177367	New	Programmer	CatB	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation Status
1191167	New	Programmer	CatB	MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result
1204882	New	Programmer	CatB	The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence
1162083	New	Programmer	CatC	16-bit T32 instruction close to breakpoint location may cause early breakpoint exception
1185469	New	Programmer	CatC	Exception packet for return stack match might return incorrect [E1:E0] field

| 1192280 | New | Programmer | CatC | IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level |
| 1207839 | New | Programmer | CatC | Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error |

16-May-2018: Changes in document version 1.0				
ID	Status	Area	Cat	Summary of erratum
1148171	New	Programmer	CatC	ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array
1151664	New	Programmer	CatC	Direct access to internal memory for L2 TLB might not update IDATAn_EL3 registers

Errata summary table

The errata associated with this product affect product versions as below.

ID	Cat	Summary	Found in versions	Fixed in version
1316063	CatA (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-write ordering violation	r0p0, r1p0	r1p1
1160841	CatB	Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock	r0p0	r1p0
1177367	CatB	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation Status	r0p0	r1p0
1191167	CatB	MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result	r0p0	r1p0
1204882	CatB	The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence	r0p0	r1p0
1220737	CatB	Streaming store under specific conditions might cause deadlock or data corruption	r0p0	r1p0
1253791	CatB	Multiple floating-point divides/ square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption Status	r0p0	r1p0
1262841	CatB	Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation	r0p0	r1p0
1273521	CatB	A T32 instruction inside an IT block followed by mispredicted speculative instruction stream might cause a deadlock	r0p0	r1p0
1450698	CatB	Software Step might prevent interrupt recognition	r0p0, r1p0	r1p1
1467687	CatB	Branch prediction for an ERET cached in the instruction cache might cause a deadlock	r0p0, r1p0	r1p1
1508412	CatB	NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock	r0p0, r1p0	r1p1
1515815	CatB	The core might execute multiple instructions before taking a software step exception or halt step exception when the executing	r0p0, r1p0	r1p1

ID	Cat	Summary	Found in versions	Fixed in version
		instruction resides in the L0 Macro-op cache		
1800714	CatB	A transient single-bit ECC error in the MMU TC RAM might lead to stale translation in the L2 TLB	r0p0, r1p0, r1p1	Open
1923201	CatB	External debugger access to debug registers might not work during Warm reset	r0p0, r1p0, r1p1	Open
1925769	CatB	Store operation that encounters multiple hits in the TLB can access regions of memory with attributes that could not be accessed at that exception level or security state	r0p0, r1p0, r1p1	Open
1286809	CatB (rare)	Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation	r0p0, r1p0	r1p1
1418842	CatB (rare)	MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data	r0p0, r1p0	r1p1
1542418	CatB (rare)	The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches	r0p0, r1p0	r1p1
1148171	CatC	ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array	r0p0	r1p0
1151664	CatC	Direct access to internal memory for L2 TLB might not update IDATAn_EL3 registers	r0p0	r1p0
1162083	CatC	16-bit T32 instruction close to breakpoint location may cause early breakpoint exception	r0p0	r1p0
1185469	CatC	Exception packet for return stack match might return incorrect [E1:E0] field	r0p0	r1p0
1192280	CatC	IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level	r0p0	r1p0
1207839	CatC	Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error	r0p0	r1p0
1220404	CatC	Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry	r0p0	r1p0
1220843	CatC	ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported	r0p0	r1p0
1244986	CatC	Illegal return event might corrupt PSTATE.UA0	r0p0	r1p0
1256789	CatC	Halting step might see extra instruction executed for some	r0p0	r1p0

ID	Cat	Summary	Found in versions	Fixed in version
		loads when crossed with snoop invalidation or ECC error		
1262908	CatC	Write-Back load after two Device-nG* stores to the same physical address might get invalid data	r0p0	r1p0
1328683	CatC	Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError	r0p0, r1p0	r1p1
1346768	CatC	TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0	r0p0, r1p0, r1p1	Open
1355135	CatC	L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events	r0p0, r1p0	r1p1
1395535	CatC	Read from PMCCNTR in AArch32 might return corrupted data	r0p0, r1p0	r1p1
1405548	CatC	MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit	r0p0, r1p0	r1p1
1415321	CatC	LDREX-STREX might succeed incorrectly when an intervening store occurs and LDREX detects a single-bit ECC error on the cache line in the L1 data cache tag RAM	r0p0, r1p0	r1p1
1421023	CatC	Portions of the branch target address recorded in ETM trace information are incorrect for an indirect branch with a malformed branch target address	r0p0, r1p0	r1p1
1487187	CatC	Waypoints from previous session might cause single-shot comparator match when trace enabled	r0p0, r1p0	r1p1
1488613	CatC	An unaligned load might initiate a prefetch request which crosses a page boundary	r0p0, r1p0	r1p1
1491015	CatC	TRCIDR3.CCITMIN value is incorrect	r0p0, r1p0	r1p1
1514033	CatC	Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE	r0p0, r1p0	r1p1
1519163	CatC	AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle	r0p0, r1p0	r1p1
1522097	CatC	The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error	r0p0, r1p0	r1p1
1523503	CatC	CPUECTLR_EL1 controls for the MMU have no affect	r0p0, r1p0	r1p1

ID	Cat	Summary	Found in versions	Fixed in version
1610369	CatC	ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect	r0p0, r1p0, r1p1	Open
1624431	CatC	CPUAMEVTYPER4_EL0 register cannot be written	r0p0, r1p0	r1p1
1662411	CatC	Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock	r0p0, r1p0, r1p1	Open
1702492	CatC	The core might not update IDATA*_EL3 correctly by a direct memory access to L1 Instruction Cache Tag or L1 Instruction TLB	r0p0, r1p0, r1p1	Open
1788065	CatC	Possible loss of CTI event	r0p0, r1p0, r1p1	Open
1788067	CatC	Loss of CTI events during warm reset	r0p0, r1p0, r1p1	Open
1827134	CatC	External debug accesses in memory access mode with SCTLR_ELx.IESB set might result in unpredictable behavior	r0p0, r1p0, r1p1	Open
1830646	CatC	Watchpoint Exception on DC ZVA does not report correct address in FAR or EDWAR	r0p0, r1p0, r1p1	Open
1857204	CatC	A memory mapped write to PMSSRR might falsely cause some PMU counters and counter overflow status to be reset after snapshot capture and read might return unknown/written data	r0p0, r1p0, r1p1	Open
1869877	CatC	ERR0MISC0_EL1.SUBARRAY, ERR0STATUS.CE and ERR0STATUS.DE values for ECC errors in the L1 data cache might be incorrect	r0p0, r1p0, r1p1	Open
1880115	CatC	Noncompliance with prioritization of Exception Catch debug events	r0p0, r1p0, r1p1	Open
1884878	CatC	The core might report incorrect fetch address to FAR_ELn when the core is fetching an instruction from a virtual address associated with a page table entry which has been modified	r0p0, r1p0, r1p1	Open
1899210	CatC	Some corrected errors might incorrectly increment ERR0MISC0.CECR or ERR0MISC0.CECO	r0p0, r1p0, r1p1	Open
1899434	CatC	PFG duplicate reported faults through a warm reset	r0p0, r1p0, r1p1	Open
1923198	CatC	IDATAn_EL3 might represent incorrect value after direct memory access to internal memory for Instruction TLB	r0p0, r1p0, r1p1	Open

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

1316063

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

Status

Fault Type: Programmer Category A (Rare)
Fault Status: Present in r0p0 and r1p0. Fixed in r1p1.

Description

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

Configurations Affected

The erratum affects all multi-core configurations.

Conditions.

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

Implications

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

Workaround

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

Category B

1160841

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

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

Set CPUACTLR2_EL1[0] to 1 and CPUACTLR2_EL1[15] to 1.

1177367

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1191167

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workarounds

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

1204882

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided if software sets CPUACTLR2_EL1 bit[11] to 0b1.

1220737

Streaming store under specific conditions might cause deadlock or data corruption

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1253791

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1262841

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

Status

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

Description

Under specific conditions, an incorrect virtual to physical mapping might happen because the L2 TLB is corrupted. The L2 TLB might be corrupted because of both:

- A translation access hitting an entry in the L2 TLB which was previously allocated by the MMU hardware prefetch mechanism.
- A TLBI VAAE1 or TLBI VAALE1 for a non-active VMID context invalidating an entry.

A subsequent translation which hits against the corrupted entry generates an incorrect translation.

Configurations Affected

This erratum affects all configurations.

Conditions

1. Virtualization is enabled.
2. The MMU hardware prefetcher installs an entry in the L2 TLB.
3. A translation request in the current VMID context hits on the prefetched entry.
4. TLBI VAAE1 or TLBI VAALE1 targeting a page within a non-active VMID context is in the process of invalidating a page.

Implications

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

Workaround

This erratum can be avoided by setting CPUECTLR_EL1[51] to 1, which disables the MMU hardware prefetcher. Setting this bit might have a small impact on performance.

1273521

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1 to increase the mispredict to fetch latency, which will have some impact on performance.

1450698

Software Step might prevent interrupt recognition

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions:

Case A:

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

Case B:

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

Implications

Case A:

Arm believes that for this product, MDSCR_EL1.KDE is not set to 1 by deployed devices in the field and is only used when debugging the system software during initial product development. In these cases, the effect of the erratum is for interrupts to be disabled even after switching to other software contexts that are not being debugged as part of the system software debugging. Arm believes that a workaround does not need to be deployed for the situation where MDSCR_EL1.KDE==1, and a workaround is not available.

Some devices are expected to run an operating system at EL2 with HCR_EL2.E2H set to 1. The implication of this erratum for such a system is that single-stepping of an untrusted user application at EL0 can lead to subsequent execution not recognizing interrupts where it should, leading to errant behavior. The software workaround described below can be deployed in the operating system at EL2 to prevent single-stepping of untrusted user applications from triggering this erratum.

Case B:

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

Workaround

When Software Step is used to debug an application under an operating system running at EL2 with HCR_EL2.E2H set to 1, the software workaround involves explicitly triggering a Software Step exception with modifications to the system call exception handler code and Software Step exception handler code. This entails setting MDSCR_EL1.KDE and MDSCR_EL1.SS and clearing PSTATE.D to trigger a Software Step exception from the system call handler. The Software Step handler then sets SPSR_ELx.D before returning back to the system call handler, where MDSCR_EL1.KDE and MDSCR_EL1.SS are restored to their original values.

If a workaround is required when MDSCR_EL1.KDE is set to 1, then please contact Arm.

1467687

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by preventing the caching of the ERET. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers:

```
LDR  x0,=0x3
MSR  S3_6_c15_c8_0,x0
LDR  x0,=0xF3D08000
MSR  S3_6_c15_c8_2,x0
LDR  x0,=0xFFF0F0FF
MSR  S3_6_c15_c8_3,x0
LDR  x0,=0x80000002003FF
MSR  S3_6_c15_c8_1,x0
ISB
```

1508412

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

Status

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

Description

Under certain conditions, execution of either a load to device or non-cacheable memory, and either a store exclusive or register read of the Physical Address Register (PAR_EL1) in close proximity might lead to a deadlock.

Configurations Affected

This erratum affects all configurations.

Conditions

Case A

In program order:

1. The core executes a store-exclusive or register read of PAR_EL1.
2. The core executes any load with device or non-cacheable memory attributes.

Case B

In program order:

1. The core executes any load with device memory attributes.
2. The core executes a store-exclusive or register read of PAR_EL1.

Implications

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

Workaround

The following workaround is to prevent an EL0 attack of the device. There is no workaround that will prevent malicious code at EL1 or higher from being able to exploit this erratum.

Case A of erratum can be avoided with the following steps:

1. Modify the software running at EL1 and above to include a DMB SY before and after accessing PAR_EL1. PAR_EL1 is not accessible from EL0.
2. Use the following write sequence to several IMPLEMENTATION DEFINED registers to have the hardware insert a DMB SY after all load-exclusive and store-exclusive instructions. The code sequence applies to r1p0 hardware and should be applied early in the boot sequence prior to any of the possible errata conditions being met:

```
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8400000
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00ff600000
MSR S3_6_c15_c8_3,x0
LDR x0,=0x00e8c00080
MSR S3_6_c15_c8_4,x0
LDR x0,=0x00ffe000c0
MSR S3_6_c15_c8_5,x0
```

```
LDR x0,=0x04004003FF
MSR S3_6_c15_c8_1,x0
ISB
```

Note that if this workaround needs to be implemented on r0p0 hardware, then the following code sequence should be used instead of the above:

```
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8400000
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00ffe00000
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4004003FF
MSR S3_6_c15_c8_1,x0
LDR x0,=0x1
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8c00040
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00ffe00040
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4004003FF
MSR S3_6_c15_c8_1,x0
ISB
```

In addition to the previous steps 1 and 2, Case B of the erratum also requires:
   3. Prevent EL0 code from accessing a location mapped with device memory attributes.

Adding DMB after load with device memory attributes for Case-B is not expected to be required as explicit accesses between ldxr/stxr is unrealistic.

One implication of this workaround is that minor performance degradation might be observed in code utilizing load-exclusive and store-exclusive instructions.

1515815

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1800714

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

Status

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

Description

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

Configurations affected

All configurations are affected.

Conditions

All of the following conditions must be met:

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

Implications

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

Workaround

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

1923201

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

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

Description

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

Configurations Affected

All configurations are affected.

Conditions

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

Implications

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

Workaround

There is no workaround.

1925769

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

Category B (rare)

1286809

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

Status

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

Description

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

Configurations Affected

The erratum affects all multi-core configurations.

Conditions

1. Core A speculatively executes a load (LD2) ahead of an older load (LD1) to the same cacheable virtual address.

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

Implications

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

Workaround

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

1418842

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

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

Implications

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

Workarounds

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

- CNTP_CTL.
- CNTP_CVAL.
- CNTP_TVAL.
- CNTV_CTL.
- CNTV_CVAL.
- CNTV_TVAL.
- CNTPCT.
- CNTVCT.
- CNTFRQ.

1542418

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

Status

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

Description

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

Configurations Affected

This erratum affects all multi-core configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by invalidating branch history before reusing any ASID for a new address space. This can be done by ensuring 60 ASIDs are selected before any ASID is reused.

Category C

1148171

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

Status

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

Description

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

Configurations affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.

1151664

Direct access to internal memory for L2 TLB might not update IDATAn_EL3 registers

Status

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

Description

Direct access to internal memory for the L2 TLB might not update the IDATAn_EL3 registers.

Configurations Affected

This erratum affects all configurations.

Conditions

> 1. The RAMINDEX register is updated to initiate a direct access to the L2 TLB.
> 2. The instruction fetch unit is not processing any snoop requests.

Implications

Direct access to internal memory is a debug feature for reading contents of certain internal memories through IMPLEMENTATION DEFINED system registers. If the above conditions are met, then any direct access to the L2 TLB memory returns invalid data. The regional clock gating prevents update of the IDATAn_EL3 register in the scenario.

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[14] to 1 for performing direct access to internal memory.

1162083

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

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1185469

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1192280

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

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround.

1207839

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.

1220404

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1220843

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs if one of the following conditions is true:

- The core issues a copyback transaction (WriteBackFull, WriteEvictFull, Evict, or WriteNoSnpFull) which then receives an error response.
- The core receives data containing an error (Poison or DErr response), but the core caches cannot defer the error by marking the data as poisoned in its caches. This occurs when the core is configured with CORE_CACHE_PROTECTION set to FALSE, or when ERR0CTLR.ED is 0.

Implications

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

Workaround

There is no workaround for this erratum.

1244986

Illegal return event might corrupt PSTATE.UA0

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

No workaround is required for this erratum.

1256789

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.

1262908

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.

1328683

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1346768

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

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

The TLBI will execute locally and broadcast with an ASID that is out of range for this configuration.

Workaround

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

1355135

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

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

Implications

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

Workaround

There is no workaround for this erratum.

1395535

Read from PMCCNTR in AArch32 might return corrupted data

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum is not expected to require a workaround.

1405548

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1415321

LDREX-STREX might succeed incorrectly when an intervening store occurs and LDREX detects a single-bit ECC error on the cache line in the L1 data cache tag RAM

Status

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

Description

If a core:

1. Detects a false miss due to a single-bit tag ECC error in the L1 data cache tag RAM on an LDREX.
2. Completes the LDREX by forwarding data from a prior store and that store is able to merge its data to the cache prior to a snoop targeting the same line, where the snoop is ordered ahead of the miss request from the load.

Then it might lead to the STREX succeeding even though there is an intervening store.

Configurations Affected

The erratum affects all multicore configurations with CORE_CACHE_PROTECTION= 1.

Conditions

1. Core A has a cache line X resident in the L1 data cache with write permissions and has one or more stores in flight.
2. Core A performs an LDREX as part of a sequence to acquire a lock. The LDREX encounters a tag single-bit ECC error, which makes the line appear as a miss.
3. The LDREX allocates a miss request buffer, but is able to forward from the older store and complete. As a result the exclusive monitor is armed in Core A, and is tracking the outstanding miss request.
4. The older store drains to the cache as the line is still in the L1 data cache.
5. Core B sends a snoop for line X and the snoop is ordered ahead of the miss request from the load. Core A responds to the snoop, but the monitor is still armed, as it was tracking the outstanding miss request.
6. Core B performs a store to the line X.
7. Core A then receives the line X on behalf of its miss request and allocates the line.
8. STREX completes successfully as the monitor is armed.

Implications

If the above conditions are met, then the STREX can succeed, even though there was an intervening store in the middle of the LDREX-STREX sequence.

Workaround

This erratum can be avoided by setting CPUACTLR3_EL1[57].

1421023

Portions of the branch target address recorded in ETM trace information are incorrect for an indirect branch with a malformed branch target address

Status

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

Description

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

Information recorded in the ETM trace buffer for branch instructions includes the Virtual Address (VA) of the branch target. An indirect branch has a malformed branch target address when either the lowermost bits of the target address stipulate a misaligned instruction address, or the uppermost bits are non-canonical. Execution of an indirect branch with a malformed target address results in an Instruction Abort. ETM trace information fails to include the malformed target address information for the branch execution, but correctly includes this information when reporting exception information for the Instruction Abort. Only the upper and lower portions of the ETM branch target VA are erroneous, by nature of excluding the malformed address information.

Configurations Affected

This erratum affects all configurations.

Conditions:

1. ETM is enabled.
2. An indirect branch with a malformed branch target address is executed and traced.

Implications

If the above conditions are met, the indirect branch with malformed target address will not include the malformed information in the branch target address in the ETM trace buffer.

Workaround

No workaround is required. The programming error should be evident to users from the ETM trace information pertaining to the resultant Instruction Abort.

1487187

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1488613

An unaligned load might initiate a prefetch request which crosses a page boundary

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1491015

TRCIDR3.CCITMIN value is incorrect

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1514033

Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions:

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

Implications

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

Workaround

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

1519163

AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle

Status

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

Description

Increments of count 16 results in an inaccurate accumulation of event INST_RETIRED (event number 0x008), which counts instructions that are architecturally executed. All other counts less than 16 are correctly captured, and the counter is incremented properly.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core enables the AMU.
2. The core enables counting on AMEVCNTR2_EL0.
3. 16 instructions retire in the same cycle, comprising 8 fused instruction-pairs (where 2 instructions are fused into a single entity for processing in the CPU).

Reads of AMEVCNTR2_EL0 give inaccurate counts, as the counter does not increment when 16 instructions retire in the same cycle.

Implications

In the unlikely event of the erratum occurring, the inaccurate counts can give a lower than expected view of the instructions being retired on the core to any software profiling this activity.

Workaround

To workaround this issue, NOP elimination must be disabled. This is done by setting both CPUACTLR_EL1[28] and CPUACTLR_EL1[26] to 1'b1. The performance delta of this workaround is expected to be small.

1522097

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

Status

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

Description

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

Configurations Affected

This erratum affects the configuration with CORE_CACHE_PROTECTION = 1.

Conditions

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

Implications

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

Workaround

This erratum has no workaround.

1523503

CPUECTLR_EL1 controls for the MMU have no affect

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions:

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

Implications

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

Workaround

There is no workaround.

1610369

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

Status

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

Description

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

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.

1624431

CPUAMEVTYPER4_EL0 register cannot be written

Status

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

Description

The AMU activity monitor counter 4 is documented as programmable at EL3, allowing users to select between two different event types. In order to select event type 0xF2, "Max Power Mitigation Mechanism", the appropriate event number must be selected by writing it to the CPUAMEVTYPER4_EL0 register. However, attempts to write the CPUAMEVTYPER4_EL0 register at EL3 result in an UNDEFINED trap.

Configurations Affected

This erratum affects all configurations.

Conditions

　　　　1.　MSR instruction attempts to write CPUAMEVTYPER4_EL0 register at EL3.

Implications

　The AMU activity counter 4 is only able to count the event 0xF1 ("High Activity").

Workaround

A workaround is not expected to be required. If you require access to the "Max Power Mitigation Mechanism" event type, then contact Arm support for more details.

1662411

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

Status

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

Description

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

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions

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

Implications

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

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

Workaround

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

1702492

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

When one of the following conditions is satisfied, the CPU can produce this erratum:

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

Implications

IDATA*_EL3 might not be updated after the completion of the direct memory access. IDATA*_EL3 might hold either an old value for L1 Instruction Cache Tag access or a corrupted value for L1 Instruction TLB access.

Workaround

This erratum has no workaround.

1788065

Possible loss of CTI event

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

CTI events might be dropped.

Workaround

This erratum has no workaround.

1788067

Loss of CTI events during warm reset

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum has no workaround.

1827134

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

1830646

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications:

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

Workaround:

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

1857204

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

Status

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

Description:

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

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

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

Workaround

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

1869877

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

Status

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

Description

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

Configurations affected

This erratum affects configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.

1880115

Noncompliance with prioritization of Exception Catch debug events

Status

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

1884878

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

Fault Type: Programmer Category C

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

Description

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

Configurations Affected

All configurations are affected.

Conditions

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

Implications

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

Workaround

There is no workaround.

1899210

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

No workaround is expected to be required.

1899434

PFG duplicate reported faults through a Warm reset

Status

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

Description

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

Configurations affected

All configurations are affected.

Conditions

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

Implications

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

Workaround

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

1923198

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

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround.