Arm Neoverse N1 (MP050)
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.
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.

Product Status

The information in this document is for a product in development and is not final.
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-885747.
  • 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

## INTRODUCTION

## ERRATA SUMMARY TABLE

<table>
<thead>
<tr>
<th>Errata ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1315703</td>
<td>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</td>
</tr>
<tr>
<td>905797</td>
<td>Failure to enforce read-after-read ordering rules</td>
</tr>
<tr>
<td>925373</td>
<td>Executing a WFx instruction while SPE is enabled might cause deadlock</td>
</tr>
<tr>
<td>931711</td>
<td>Reads from DSU CLUSTER* or ERX* system registers might return corrupted data</td>
</tr>
<tr>
<td>977072</td>
<td>Accessing certain Debug or Generic Timer system registers in AArch32 might cause incorrect system register values</td>
</tr>
<tr>
<td>981980</td>
<td>Interrupt is taken immediately after MSR DAIF instruction masks the interrupt</td>
</tr>
<tr>
<td>1039219</td>
<td>When using SPE, sampling certain system register instructions might cause deadlock</td>
</tr>
<tr>
<td>1043202</td>
<td>AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check</td>
</tr>
<tr>
<td>1073348</td>
<td>Concurrent instruction TLB miss and mispredicted return instruction might fetch wrong instruction stream</td>
</tr>
<tr>
<td>1130799</td>
<td>TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data</td>
</tr>
<tr>
<td>1165347</td>
<td>Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock</td>
</tr>
<tr>
<td>1165522</td>
<td>Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation</td>
</tr>
<tr>
<td>1188873</td>
<td>MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result</td>
</tr>
<tr>
<td>1207823</td>
<td>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</td>
</tr>
<tr>
<td>1220197</td>
<td>Streaming store under specific conditions might cause deadlock or data corruption</td>
</tr>
<tr>
<td>1257314</td>
<td>Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption</td>
</tr>
<tr>
<td>1262606</td>
<td>Concurrent instruction TLB miss and mispredicted branch instruction located at the end of 32MB region might fetch wrong instruction stream</td>
</tr>
<tr>
<td>1262888</td>
<td>Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation</td>
</tr>
<tr>
<td>1275112</td>
<td>A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock</td>
</tr>
<tr>
<td>1354823</td>
<td>SnpOnceFwd might return incorrect data</td>
</tr>
<tr>
<td>1467587</td>
<td>HCR_EL2.TOCU incorrectly applies during EL0 execution when HCR_EL2. (E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1</td>
</tr>
<tr>
<td>1533195</td>
<td>Accessing a memory location using mismatched shareability attributes might cause loss of coherency</td>
</tr>
<tr>
<td>1688567</td>
<td>Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation</td>
</tr>
<tr>
<td>1688568</td>
<td>Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer</td>
</tr>
<tr>
<td>1286807</td>
<td>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</td>
</tr>
<tr>
<td>1418040</td>
<td>MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data</td>
</tr>
</tbody>
</table>
1542419 The core might fetch a stale instruction from memory which violates the ordering of instruction fetches 50
901361 Failure to report or incorrect reporting of L2 data RAM ECC errors 51
901865 Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock 52
902290 Persistent error response to transactions issued on behalf of Page descriptor Access bit and Dirty bit updates might livelock 53
909055 Failure to record L1 data cache access event when using the SPE 54
930017 Failure to sign-extend instruction virtual address when using the SPE 55
933092 Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort 56
933779 DBGDTRTX register fails to hold value through Warm reset 57
934968 DCPSx instruction with SCTLR_EL1.IESB = 1 while in debug state might not execute correctly 58
937437 An SPE buffer full event might clear PMBSR_EL1.DL and PMBSR_EL1.EA 59
941868 Deferred errors might cause silent data corruption following a hardware update of Access and Dirty bits in a translation table entry 60
944783 Address breakpoint might cause a deadlock with certain AArch32 T32 code sequences 61
961111 L2 might report multiple RAS errors for the same prefetch request 62
964384 Stuck-at-fault in L1 instruction cache data array might cause deadlock with certain AArch32 T32 code sequences 63
978245 Executing unallocated encoding in conversion between floating-point and integer instruction class does not generate Undefined Instruction exception 64
986709 MRS to DBGDTR_EL0 might cause EDSCR.RXfull bit to clear incorrectly 65
988575 Unaligned cache line split load to NC or Device memory, tagged with poison or external error on its first half, might cause data corruption 66
1051464 CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed 67
1057923 Extra instruction might be executed during Halting Step when stepping WFI, WFE and some self-synchronizing system register writes 68
1069401 Debug APB accesses to the ELA RAM might return incorrect data 69
1096402 Exception packet for return stack match might return incorrect [E1:E0] field 70
1109624 Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock 71
1119735 16-bit T32 instruction close to breakpoint location might cause early breakpoint exception 72
1126105 Read from L1 instruction cache data array using RAMINDEX operation might return data from the wrong location 73
1144394 Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error 74
1192279 IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level 75
1194748 The ERXADDR_EL1 register might report an incorrect physical address for an L1 data tag RAM single-bit correctable ECC error 76
1194749 ERROMISRISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array 77
1214504 Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry 78
1227053 Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering 79
<table>
<thead>
<tr>
<th>Issue Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1227629</td>
<td>ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported</td>
</tr>
<tr>
<td>1244984</td>
<td>Illegal return event might corrupt PSTATE.UA0</td>
</tr>
<tr>
<td>1256788</td>
<td>Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error</td>
</tr>
<tr>
<td>1264383</td>
<td>Write-Back load after two Device-nG* stores to the same physical address might get invalid data</td>
</tr>
<tr>
<td>1346756</td>
<td>TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0</td>
</tr>
<tr>
<td>1349291</td>
<td>Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError</td>
</tr>
<tr>
<td>1356341</td>
<td>L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events</td>
</tr>
<tr>
<td>1395332</td>
<td>Read from PMCCNTR in AArch32 might return corrupted data</td>
</tr>
<tr>
<td>1406411</td>
<td>MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.(N,C,Z,V,GE) on debug exit</td>
</tr>
<tr>
<td>1408724</td>
<td>Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address</td>
</tr>
<tr>
<td>1415323</td>
<td>Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line</td>
</tr>
<tr>
<td>1430754</td>
<td>Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences</td>
</tr>
<tr>
<td>1487185</td>
<td>Waypoints from previous session might cause single-shot comparator match when trace enabled</td>
</tr>
<tr>
<td>1490853</td>
<td>TRCDIR3.CCITMIN value is incorrect</td>
</tr>
<tr>
<td>1514034</td>
<td>Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE</td>
</tr>
<tr>
<td>1523502</td>
<td>CPUECTL_EL1 controls for the MMU have no affect</td>
</tr>
<tr>
<td>1627784</td>
<td>ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect</td>
</tr>
<tr>
<td>1655746</td>
<td>MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption</td>
</tr>
<tr>
<td>1662732</td>
<td>Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior</td>
</tr>
<tr>
<td>1694299</td>
<td>Instruction sampling bias exists in SPE implementation</td>
</tr>
<tr>
<td>1697035</td>
<td>Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock</td>
</tr>
</tbody>
</table>
r1p0 implementation fixes

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

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

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

r2p0 implementation fixes

Note the following errata might be fixed in some implementations of r2p0. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

| REVIDR_EL1[0] | 1220197 Streaming store under specific conditions might cause deadlock or data corruption |

Note that there is no change to the MIDR_EL1 which remains at r2p0 but the REVIDR_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.

r3p1 implementation fixes

Note the following errata might be fixed in some implementations of r3p1. This can be determined by reading the REVIDR_EL1 register where a set bit indicates that the erratum is fixed in this part.

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

Note that there is no change to the MIDR_EL1 which remains at r3p1 but the REVIDR_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of MIDR_EL1 and REVIDR_EL1.
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 18 identifies errata that have been fixed in each product revision.

### 14-Feb-2020: Changes in document version 20.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1688567</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation</td>
</tr>
<tr>
<td>1688568</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer</td>
</tr>
<tr>
<td>1655746</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption</td>
</tr>
<tr>
<td>1694299</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Instruction sampling bias exists in SPE implementation</td>
</tr>
</tbody>
</table>

### 20-Dec-2019: Changes in document version 19.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1627784</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect</td>
</tr>
<tr>
<td>1662732</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior</td>
</tr>
<tr>
<td>1697035</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Executing a cache maintenance by set/way instruction targeting the L1 data cache in the presence of snoops might result in a deadlock</td>
</tr>
</tbody>
</table>

### 27-Sep-2019: Changes in document version 18.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1542419</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>The core might fetch a stale instruction from memory which violates the ordering of instruction fetches</td>
</tr>
</tbody>
</table>

### 24-Sep-2019: Changes in document version 17.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1533195</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Accessing a memory location using mismatched shareability attributes might cause loss of coherency</td>
</tr>
<tr>
<td>1542419</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>The core might fetch a stale instruction from memory which violates the ordering of instruction fetches</td>
</tr>
<tr>
<td>ID</td>
<td>Status</td>
<td>Area</td>
<td>Cat</td>
<td>Summary of erratum</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>------------</td>
<td>------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1514034</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE</td>
</tr>
<tr>
<td>1523502</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>CPUECTLR_EL1 controls for the MMU have no affect</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1467587</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>HCR_EL2.TOCU incorrectly applies during EL0 execution when HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1</td>
</tr>
<tr>
<td>1346756</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0</td>
</tr>
<tr>
<td>1487185</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Waypoints from previous session might cause single-shot comparator match when trace enabled</td>
</tr>
<tr>
<td>1490853</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>TRCIDR3.CCITMIN value is incorrect</td>
</tr>
</tbody>
</table>

30-Apr-2019: Changes in document version 15.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1354823</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>SnpOnceFwd might return incorrect data</td>
</tr>
<tr>
<td>1418040</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data</td>
</tr>
<tr>
<td>1349291</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError</td>
</tr>
<tr>
<td>1356341</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events</td>
</tr>
<tr>
<td>1395332</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Read from PMCCNTR in AArch32 might return corrupted data</td>
</tr>
<tr>
<td>1406411</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit</td>
</tr>
<tr>
<td>1408724</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address</td>
</tr>
<tr>
<td>1415323</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line</td>
</tr>
</tbody>
</table>
### 26-Mar-2019: Changes in document version 14.0

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

### 08-Mar-2019: Changes in document version 13.0

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

### 21-Nov-2018: Changes in document version 12.0

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

04-Oct-2018: Changes in document version 11.0

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

**07-Sep-2018: Changes in document version 10.0**

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1165522</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation</td>
</tr>
<tr>
<td>1188873</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result</td>
</tr>
<tr>
<td>1220197</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Streaming store under specific conditions might cause deadlock or data corruption</td>
</tr>
</tbody>
</table>

**01-Aug-2018: Changes in document version 9.0**

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1130799</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data</td>
</tr>
<tr>
<td>1165347</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock</td>
</tr>
<tr>
<td>1165522</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation</td>
</tr>
<tr>
<td>1188873</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result</td>
</tr>
<tr>
<td>1207823</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>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</td>
</tr>
<tr>
<td>1096402</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Exception packet for return stack match might return incorrect [E1:E0] field</td>
</tr>
<tr>
<td>1109624</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock</td>
</tr>
<tr>
<td>1119735</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>16-bit T32 instruction close to breakpoint location might cause early breakpoint exception</td>
</tr>
<tr>
<td>1126105</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Read from L1 instruction cache data array using RAMINDEX operation might return data from the wrong location</td>
</tr>
</tbody>
</table>
### 1144394
- **Updated**
- **Programmer**
- **CatC**
  - **Summary of erratum**
  - Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error

### 1192279
- **New**
- **Programmer**
- **CatC**
  - IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level

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

### 1194749
- **Updated**
- **Programmer**
- **CatC**
  - ERR0MISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array

### 1214504
- **New**
- **Programmer**
- **CatC**
  - Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry

### 1227629
- **New**
- **Programmer**
- **CatC**
  - ERR0STATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported

### 22-Jun-2018: Changes in document version 8.0

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

### 13-Apr-2018: Changes in document version 7.0

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

03-Apr-2018: Changes in document version 6.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1043202</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td></td>
</tr>
<tr>
<td>1051464</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td></td>
</tr>
</tbody>
</table>

09-Feb-2018: Changes in document version 5.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1039219</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>When using SPE, sampling certain system register instructions might cause deadlock</td>
</tr>
<tr>
<td>1043202</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check</td>
</tr>
<tr>
<td>1051464</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed</td>
</tr>
</tbody>
</table>


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

**25-Sep-2017: Changes in document version 3.0**

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td>931711</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Reads from DSU CLUSTER* or ERX* system registers might return corrupted data</td>
</tr>
<tr>
<td>933092</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort</td>
</tr>
<tr>
<td>933779</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>DBGDTRTX register fails to hold value through Warm reset</td>
</tr>
<tr>
<td>934968</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>DCPSx instruction with SCTLR_EL1.IESB = 1 while in debug state might not execute correctly</td>
</tr>
</tbody>
</table>

**18-Aug-2017: Changes in document version 2.0**

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

**29-May-2017: Changes in document version 1.0**

<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Area</th>
<th>Cat</th>
<th>Summary of erratum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No errata in this document version.</td>
</tr>
</tbody>
</table>
## Errata summary table

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

<table>
<thead>
<tr>
<th>ID</th>
<th>Cat</th>
<th>Summary</th>
<th>Found in versions</th>
<th>Fixed in version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1315703</td>
<td>CatA</td>
<td>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</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>905797</td>
<td>CatB</td>
<td>Failure to enforce read-after-read ordering rules</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>925373</td>
<td>CatB</td>
<td>Executing a WFx instruction while SPE is enabled might cause deadlock</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>931711</td>
<td>CatB</td>
<td>Reads from DSU CLUSTER* or ERX* system registers might return corrupted data</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>977072</td>
<td>CatB</td>
<td>Accessing certain Debug or Generic Timer system registers in AArch32 might cause incorrect system register values</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>981980</td>
<td>CatB</td>
<td>Interrupt is taken immediately after MSR DAIF instruction masks the interrupt</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1039219</td>
<td>CatB</td>
<td>When using SPE, sampling certain system register instructions might cause deadlock</td>
<td>r0p0, r1p0</td>
<td>r2p0</td>
</tr>
<tr>
<td>1043202</td>
<td>CatB</td>
<td>AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check</td>
<td>r0p0, r1p0</td>
<td>r2p0</td>
</tr>
<tr>
<td>1073348</td>
<td>CatB</td>
<td>Concurrent instruction TLB miss and mispredicted return instruction might fetch wrong instruction stream</td>
<td>r0p0, r1p0</td>
<td>r2p0</td>
</tr>
<tr>
<td>1130799</td>
<td>CatB</td>
<td>TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td>1165347</td>
<td>CatB</td>
<td>Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td>1165522</td>
<td>CatB</td>
<td>Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td>1188873</td>
<td>CatB</td>
<td>MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td>1207823</td>
<td>CatB</td>
<td>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</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td>ID</td>
<td>Cat</td>
<td>Summary</td>
<td>Found in versions</td>
<td>Fixed in version</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1220197</td>
<td>CatB</td>
<td>Streaming store under specific conditions might cause deadlock or data corruption</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td>1257314</td>
<td>CatB</td>
<td>Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1262606</td>
<td>CatB</td>
<td>Concurrent instruction TLB miss and mispredicted branch instruction located at the end of 32MB region might fetch wrong instruction stream</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1262888</td>
<td>CatB</td>
<td>Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1275112</td>
<td>CatB</td>
<td>A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1354823</td>
<td>CatB</td>
<td>SnpOnceFwd might return incorrect data</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1467587</td>
<td>CatB</td>
<td>HCR_EL2.TOCU incorrectly applies during EL0 execution when HCR_EL2.(E2H,TGE)=(1,1), SCTLR_EL1.UCI=1, and SCTLR_EL2.UCI=1</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1533195</td>
<td>CatB</td>
<td>Accessing a memory location using mismatched shareability attributes might cause loss of coherency</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1688567</td>
<td>CatB</td>
<td>Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1688568</td>
<td>CatB</td>
<td>Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1286807</td>
<td>CatB (rare)</td>
<td>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</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1418040</td>
<td>CatB (rare)</td>
<td>MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1542419</td>
<td>CatB (rare)</td>
<td>The core might fetch a stale instruction from memory which violates the ordering of instruction fetches</td>
<td>r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>ID</td>
<td>Cat</td>
<td>Summary</td>
<td>Found in versions</td>
<td>Fixed in version</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>901361</td>
<td>CatC</td>
<td>Failure to report or incorrect reporting of L2 data RAM ECC errors</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>901865</td>
<td>CatC</td>
<td>Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>902290</td>
<td>CatC</td>
<td>Persistent error response to transactions issued on behalf of Page descriptor Access bit and Dirty bit updates might livelock</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>909055</td>
<td>CatC</td>
<td>Failure to record L1 data cache access event when using the SPE</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>930017</td>
<td>CatC</td>
<td>Failure to sign-extend instruction virtual address when using the SPE</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>933092</td>
<td>CatC</td>
<td>Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>933779</td>
<td>CatC</td>
<td>DBGDTRTX register fails to hold value through Warm reset</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>934968</td>
<td>CatC</td>
<td>DCPSx instruction with SCTL_EL1.IESB = 1 while in debug state might not execute correctly</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>937437</td>
<td>CatC</td>
<td>An SPE buffer full event might clear PMBSR_EL1.DL and PMBSR_EL1.EA</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>941868</td>
<td>CatC</td>
<td>Deferred errors might cause silent data corruption following a hardware update of Access and Dirty bits in a translation table entry</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>944783</td>
<td>CatC</td>
<td>Address breakpoint might cause a deadlock with certain AArch32 T32 code sequences</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>961111</td>
<td>CatC</td>
<td>L2 might report multiple RAS errors for the same prefetch request</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>964384</td>
<td>CatC</td>
<td>Stuck-at-fault in L1 instruction cache data array might cause deadlock with certain AArch32 T32 code sequences</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>978245</td>
<td>CatC</td>
<td>Executing unallocated encoding in conversion between floating-point and integer instruction class does not generate Undefined Instruction exception</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>986709</td>
<td>CatC</td>
<td>MRS to DBGDTR_EL0 might cause EDSCR.RXfull bit to clear incorrectly</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>988575</td>
<td>CatC</td>
<td>Unaligned cache line split load to NC or Device memory, tagged with poison or external error on its first half, might cause data corruption</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1051464</td>
<td>CatC</td>
<td>CTI trigger occurring on same cycle PREADYCD is received</td>
<td>r0p0, r1p0</td>
<td>r2p0</td>
</tr>
<tr>
<td>ID</td>
<td>Cat</td>
<td>Summary</td>
<td>Found in versions</td>
<td>Fixed in version</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1057923</td>
<td>CatC</td>
<td>Extra instruction might be executed during Halting Step when stepping</td>
<td>r0p0, r1p0</td>
<td>r2p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WFI, WFE and some self-synchronizing system register writes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1069401</td>
<td>CatC</td>
<td>Debug APB accesses to the ELA RAM might return incorrect data</td>
<td>r0p0, r1p0</td>
<td>r2p0</td>
</tr>
<tr>
<td>1096402</td>
<td>CatC</td>
<td>Exception packet for return stack match might return incorrect</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[E1:E0] field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1109624</td>
<td>CatC</td>
<td>Continuous failing STREX with VA alias access outside mispredicted</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exclusive sequence (LDREX/STREX) loop might cause livelock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1119735</td>
<td>CatC</td>
<td>16-bit T32 instruction close to breakpoint location might cause</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>early breakpoint exception</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1126105</td>
<td>CatC</td>
<td>Read from L1 instruction cache data array using RAMINDEX operation</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>might return data from the wrong location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1144394</td>
<td>CatC</td>
<td>Software step might see extra instruction executed for some</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>loads when crossed with snoop invalidation or ECC error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1192279</td>
<td>CatC</td>
<td>IMPLEMENTATION DEFINED fault for unsupported atomic operations</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is not routed to proper Exception level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1194748</td>
<td>CatC</td>
<td>The ERXADDR_EL1 register might report an incorrect physical address for</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>an L1 data tag RAM single-bit correctable ECC error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1194749</td>
<td>CatC</td>
<td>ERR0MISC0 might report incorrect BANK and SUBBANK values for parity</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>errors in L1 instruction cache data array</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1214504</td>
<td>CatC</td>
<td>Direct access to L1 data TLB might report incorrect value of valid bit</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of the corresponding TLB entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1227053</td>
<td>CatC</td>
<td>Streaming writes to memory mapped Non-shareable and write-back might</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cause data corruption because of reordering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1227629</td>
<td>CatC</td>
<td>ERR0STATUS.SERR encoding is incorrect for error responses from slave</td>
<td>r0p0, r1p0, r2p0</td>
<td>r3p0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and deferred data errors from slave which are not supported</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1244984</td>
<td>CatC</td>
<td>Illegal return event might corrupt PSTATE.UA0</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1256788</td>
<td>CatC</td>
<td>Halting step might see extra instruction executed for some</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>loads when crossed with snoop invalidation or ECC error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Cat</td>
<td>Summary</td>
<td>Found in versions</td>
<td>Fixed in version</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1264383</td>
<td>CatC</td>
<td>Write-Back load after two Device-nG stores to the same physical address might get invalid data</td>
<td>r0p0, r1p0, r2p0, r3p0</td>
<td>r3p1</td>
</tr>
<tr>
<td>1346756</td>
<td>CatC</td>
<td>TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1349291</td>
<td>CatC</td>
<td>Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1356341</td>
<td>CatC</td>
<td>L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1395332</td>
<td>CatC</td>
<td>Read from PMCCNTR in AArch32 might return corrupted data</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1406411</td>
<td>CatC</td>
<td>MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.(N,C,Z,V,GE) on debug exit</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1408724</td>
<td>CatC</td>
<td>Portions of the branch target address recorded in ETM trace information might be incorrect for some branches immediately preceding an indirect branch with a malformed branch target address</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1415323</td>
<td>CatC</td>
<td>Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1430754</td>
<td>CatC</td>
<td>Write to External Debug Registers might cause a deadlock with certain AArch32 T32 code sequences</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1</td>
<td>r4p0</td>
</tr>
<tr>
<td>1487185</td>
<td>CatC</td>
<td>Waypoints from previous session might cause single-shot comparator match when trace enabled</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1490853</td>
<td>CatC</td>
<td>TRCDIDR3.CCITMIN value is incorrect</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1514034</td>
<td>CatC</td>
<td>Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1523502</td>
<td>CatC</td>
<td>CPUECTLREL1 controls for the MMU have no affect</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1627784</td>
<td>CatC</td>
<td>ERHOMISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1655746</td>
<td>CatC</td>
<td>MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1662732</td>
<td>CatC</td>
<td>Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior</td>
<td>r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>ID</td>
<td>Cat</td>
<td>Summary</td>
<td>Found in versions</td>
<td>Fixed in version</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1694299</td>
<td>CatC</td>
<td>Instruction sampling bias exists in SPE implementation</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
<tr>
<td>1697035</td>
<td>CatC</td>
<td>Executing a cache maintenance by set/way instruction targeting the</td>
<td>r0p0, r1p0, r2p0, r3p0, r3p1, r4p0</td>
<td>Open</td>
</tr>
</tbody>
</table>
Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

1315703

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

Status

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

Description

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

Configurations Affected

The erratum affects all multi-core configurations.

Conditions.

1. Core A has in program order a store (ST1) and a younger load (LD1) to the same cacheable virtual address.
2. Core B marks the associated translation table entry invalid, followed by a DSB; TLBI; DSB sequence which generates a sync request to Core A.
3. LD1 executes speculatively past ST1 and returns its result using the original physical address (PA1) under specific rare conditions before Core A has responded to the sync request.
4. At the time of receiving the sync request, on Core A:
   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 PSTATE.SSBS to 0 or CPUACTLR2_EL1[16] to 1, hence preventing LD1 from speculating past ST1. This will have a performance impact on general workloads.
Category B

905797

Failure to enforce read-after-read ordering rules

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

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

Implications

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

Workaround

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

However, for systems using an ACE interconnect or a CHI interconnect that does not support far atomic operations (thus input BROADCASTATOMIC pin is tied to 0), setting CPUACTRL2_EL1[2] to 1 will prevent the conditions necessary to hit this erratum.
925373
Executing a WFx instruction while SPE is enabled might cause deadlock

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

- Disabling SPE profiling.
- Setting CPUACTLR3_EL1[33] and CPUACTLR3_EL1[34] to 1. Setting CPUACTLR3_EL1[33] to 1 results in WFE instruction to be executed as a NOP instruction. Setting CPUACTLR3_EL1[34] to 1 results in WFI instruction to be executed as a NOP instruction.
931711
Reads from DSU CLUSTER* or ERX* system registers might return corrupted data

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

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

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

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

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

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

Workaround
In most cases, this erratum can be avoided by inserting an ISB instruction before the MRS to the CLUSTER* or ERX* system register, as the ISB can prevent the MRS from being speculatively executed.
977072

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

The erratum can be avoided by trapping MRC/MCR/MRRC/MCRR accesses in AArch32 to the affected registers and doing the equivalent code sequence in the trap handler. To trap the CNT* accesses, set CNTKCTL_EL1.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0. To trap the DBG* accesses, set MDSCR_EL1.TDCC to 1.
981980
Interrupt is taken immediately after MSR DAIF instruction masks the interrupt

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
Generally, it is expected that software will be robust against taking an interrupt immediately after masking it in PSTATE. If not, then a workaround is to subtract 4 from the ELR and clear the relevant mask bit in the SPSR when the interrupt vector is entered erroneously.
1039219
When using SPE, sampling certain system register instructions might cause deadlock

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

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

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

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

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

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

Workaround
This erratum can be avoided by disabling SPE profiling.
1043202
AArch32 T32 CLREX in an IT block will clear exclusive monitor even if it fails condition code check

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by replacing all T32 CLREX instructions with an ISB instruction. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers:

LDR x0,=0x0
MSR S3_6_c15_c8_0,x0 ; MSR CPUPSELR_EL3, X0
LDR x0,=0xF3BF8F2F
MSR S3_6_c15_c8_2,x0 ; MSR CPUPOR_EL3, X0
LDR x0,=0xFFFFFFFF
MSR S3_6_c15_c8_3,x0 ; MSR CPUPMR_EL3, X0
LDR x0,=0x800200071
MSR S3_6_c15_c8_1,x0 ; MSR CPUPCR_EL3, X0
ISB
1073348
Concurrent instruction TLB miss and mispredicted return instruction might fetch wrong instruction stream

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUACTLR_EL1[6] to 1, which disables static prediction.
1130799
TLBI VAAE1 or TLBI VAALE1 targeting a page within hardware page aggregated address translation data in the L2 TLB might cause corruption of address translation data

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. Full hardware page aggregation is enabled by setting CPUECTLRL1[48] to 1.
2. Multiple ASIDs have aggregated address translation data in different ways of the L2 TLB for the same VA range.
3. TLBI VAAE1 or TLBI VAALE1 targeting a page within the aggregated address translation data is executed.

Implications

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

Workaround

This erratum can be avoided by setting CPUACTLR2_EL1[59] to 1. Setting CPUACTLR2_EL1[59] to 1 might have a small impact on performance.
1165347
Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock

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

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

Configurations Affected
The erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUACTLR2_EL1[0] to 1 and CPUACTLR2_EL1[15] to 1.
1165522

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

When context-switching the register state for an out-of-context translation regime, system software at EL2 or above must ensure that all intermediate states during the context-switch would report a level 0 translation fault in response to an AT instruction targeting the out-of-context translation regime. Note that a workaround is only required if the system software contains an AT instruction as part of an executable page.
1188873
MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workarounds
The erratum can be avoided by trapping MRC/MCR/MRRC/MCRR accesses in AArch32 to the affected registers and doing the equivalent code sequence in the trap handler. To trap the CNT* accesses, set CNTKCTL_EL1.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0. If HCR_EL2.{E2H,TGE}={1,1} then set CNTHCTL_EL2.{EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN} to 0. The following registers will be trapped: CNTP_CTL, CNTP_CVAL, CNTP_TVAL, CNTV_CTL, CNTV_CVAL, CNTV_TVAL, CNTPCT, CNTVCT, CNTFRQ.
1207823
The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUACTLR2_EL1[11] to 1.
1220197

Streaming store under specific conditions might cause deadlock or data corruption

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by setting CPUECTLR_EL1[25:24] to 0b11, which disables write streaming to the L2. This will have an impact on performance for streaming workloads.
1257314

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

Description

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

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.
1262606

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. A branch instruction located at the end of a 32MB region (if the branch address is PC=X[63:0], PC[24:6]=all1) is speculatively executed.
3. A translation table walk response for the other 32MB region arrives around the time the branch is speculated and written into the L1 instruction TLB.
4. When the branch resolves, bits[24:0] of the speculated address must match the actual address.

Implications

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

Workaround

This erratum can be avoided by setting CPUACTLR_EL1[13] to 1, which delays instruction fetch after branch misprediction. This workaround will have a small impact on performance.
1262888
Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation

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

Description
Under specific conditions, an incorrect virtual to physical mapping might happen because the L2 TLB 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.
1275112

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. A T32 instruction is inside an IT block.
2. Subsequent instructions repeatedly create branch misprediction. Branch predictor misprediction occurs either because:
   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, which delays instruction fetch after branch misprediction. This workaround will have a small impact on performance.

The workaround for this erratum is the same as the workaround for erratum 1262606.
1354823
SnPOnceFwd might return incorrect data

Status

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

Description

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

Configurations affected

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

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

Conditions

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

Implications

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

Workaround

For systems using the CMN-600 interconnect, set por_hnf_aux_ctl[6]=1 to disable SnPOnceFwd for ReadOnce. HN-F will use SnPShared instead. ReadOnceMakeInvalid and ReadOnceCleanInvalid use SnPUnique and are not affected. The workaround might cause some performance degradation, the extent of which is highly system and workload dependent.
1467587

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by clearing HCR_EL2.TOCU when entering HCR_EL2.(E2H,TGE)=(1,1) mode.
1533195

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

Status

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

Description

If two PEs access the same memory location with mismatched shareability attributes, accesses performed by a PE using a shareable mapping might cause data corruption in a PE using a non-shareable mapping of the same physical address. Similarly, stashing into a PE whose caches were allocated using a non-shareable mapping might cause data corruption.

Configurations Affected

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_BIG_CORES</td>
<td>1</td>
</tr>
<tr>
<td>L3_CACHE</td>
<td>FALSE</td>
</tr>
<tr>
<td>ACE</td>
<td>FALSE</td>
</tr>
<tr>
<td>PORTER_SAM</td>
<td>TRUE</td>
</tr>
<tr>
<td>ACP</td>
<td>FALSE</td>
</tr>
<tr>
<td>PERIPH_PORT</td>
<td>FALSE</td>
</tr>
<tr>
<td>ASYNC_BRIDGE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Conditions:

1. PE0 accesses a memory location using cacheable and non-shareable attributes and later writes to the same location, resulting in dirty data in the PE's caches.
2. PE1 accesses the same memory location using cacheable and shareable attributes. Accesses include reads, writes, and cache maintenance operations.
3. Interconnect does not filter snoop traffic to PEs and as a result snoops PE0.
4. Snoop to PE0 causes a cache state change but does not cause a copyback of PE0's dirty data or a stash snoop to PE0 that might request data using a DataPull response, but internal queues are not enabled to expect data.

Implications

If the above conditions are met, PE0 might experience data corruption.

Workaround

Avoid using mismatched shareability attributes for aliases of the same memory location.
1688567

Hardware management of dirty state and the Access flag by SPE might fail, resulting in an unsupported FSC code and incorrect EC code in PMBSR_EL1 on a buffer translation

Status

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

Description

When Stage 2 dirty and access flag updates are turned off, a failed profiling buffer translation request might result in reporting a Stage 2 Data Abort code in PMBSR_EL1.EC. This also results in an Unsupported Exclusive or Atomic Access fault status code update in PMBSR_EL1, which is not one of the defined FSC codes for this register.

Configurations Affected

This erratum affects all configurations.

Conditions

SPE is enabled and the following conditions are true:

1. Hardware Management of dirty state and access flag update in Stage 1 translations is enabled in TCR_EL1.
2. Hardware Management of dirty state and access flag update in Stage 2 translations is disabled.

Implications

There might be a loss of sampling data as software needs to restart the profiling session to recover from this error.

Workaround

This erratum can be avoided by pre-dirtying the SPE buffer pages.
1688568
Enabling SPE might result in a speculative update of the translation table descriptor of the page following the Statistical Profiling Buffer

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

Category B (rare)

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

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

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

Configurations Affected
The erratum affects all multi-core configurations.

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

Implications

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

Workaround

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.
1418040

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

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

Implications

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

Workarounds

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

- CNTP_CTL.
- CNTP_CVAL.
- CNTP_TVAL.
- CNTV_CTL.
- CNTV_CVAL.
- CNTV_TVAL.
- CNTPCT.
- CNTVCT.
- CNTFRQ.
1542419
The core might fetch a stale instruction from memory which violates the ordering of instruction fetches

Status

Fault Type: Programmer Category B Rare
Fault Status: Present in r3p0, r3p1, and r4p0. Open

Description

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

Configurations Affected

This erratum affects all configurations with

| COHERENT_ICACHE | TRUE |

Conditions

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

Implications

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

Workaround

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

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

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

```asm
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,=0xEE670D35
MSR S3_6_c15_c8_2,x0
LDR x0,= 0xFFFF0FFF
```
The above indicates an IMPLEMENTATION DEFINED exception to EL3 with the ESR_EL3.EC set to 0b11111 (0x1F).

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

Category C

901361

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

If this erratum occurs, either:

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

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

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

Workaround

A partial workaround is possible. Setting CPUACTLR2_EL1[45] to 1 forces reporting and logging of all L2 data RAM ECC errors. However, the reported physical address might still be incorrect. Other recorded information, such as array, subarray, and index are correct. Setting CPUACTLR2_EL1[45] to 1 might have a small impact on performance.
901865  
Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum is not expected to be encountered in real software. There is no workaround for this erratum.
902290
Persistent error response to transactions issued on behalf of Page descriptor Access bit and Dirty bit updates might livelock

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
There is no workaround for this erratum.
**909055**

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

**Status**

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

**Description**

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

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

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

**Implications**

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

**Workaround**

There is no workaround for this erratum.
930017

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

Software can examine instruction virtual address bit[48] to determine the proper value for all bits in the range [55:48]. If bit[48] is equal to 0, then all bits in the range should be read as 0. If bit[48] is equal to 1, then all bits in the range should be read as 1.
933092
Critical beat data for an L2 cache miss, poisoned or tagged with error, consumed by a load without reporting an abort

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

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

Configurations Affected
The erratum affects all configurations.

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

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

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

If a workaround is required, set CPUACTLR2_EL1[43] to 1. This prevents critical beat forwarding, and ensures that the load will abort in case the system reports an error. Setting CPUACTLR2_EL1[43] to 1 might have a small impact on performance.
**933779**

**DBGDTRTX register fails to hold value through Warm reset**

**Status**

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

**Description**

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

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

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

**Implications**

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

**Workaround**

There is no workaround for this erratum.
934968
DCPSx instruction with SCLTR_EL1.IESB = 1 while in debug state might not execute correctly

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

Description
A DCPS1, DCPS2, or DCPS3 instruction that is executed in debug state while SCLTR_EL1.IESB is set to 1 might result in incorrect execution of the instruction.

Configurations Affected
This erratum affects all configurations.

Conditions
1. SCLTR_EL1.IESB is set to 1.
2. The core is in debug state.
3. The core executes a DCPS1, DCPS2, or DCPS3 instruction.

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

Workaround
The erratum can be avoided by clearing SCLTR_EL1.IESB before executing a DCPSx instruction in debug state. If the core is in a state where SCLTR_EL1 writes are trapped, then up to three write attempts might be required where each attempt might be trapped to a higher Exception level.
937437
An SPE buffer full event might clear PMBSR_EL1.DL and PMBSR_EL1.EA

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

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

Configurations Affected
This erratum affects all configurations.

Conditions
There are two cases in which the erratum might occur.

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

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

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

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

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

Workaround
There is no workaround for this erratum.
941868
Deferred errors might cause silent data corruption following a hardware update of Access and Dirty bits in a translation table entry

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
If the system supports far atomic accesses to cacheable memory, then setting CPUACTLR2_EL1[41] to 1 forces the L2 cache to perform hardware updates of the Access and Dirty flags as far atomics.
944783

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The breakpoint is set on a Cacheable line.
3. The breakpoint is not quadword aligned.
4. The cache line contains at least two 32-bit instructions, of which at least one must be after the breakpoint.

Implications

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

Workaround

Any interrupt will break the processor out of the deadlock state. The deadlock can be avoided by forcing the page containing the T32 instruction to be Non-cacheable.
961111
L2 might report multiple RAS errors for the same prefetch request

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
No workaround is required for this erratum.
964384

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

Any interrupt will break the processor out of the deadlock state. The stuck-at-fault can be bypassed by forcing the page containing the T32 instruction to be Non-cacheable.
978245

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
986709
MRS to DBGDTR_EL0 might cause EDSCR.RXfull bit to clear incorrectly

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

Workaround
This erratum can be avoided by inserting an ISB instruction before the MRS to DBGDTR_EL0, because the ISB can prevent the MRS from being speculatively executed.
988575
Unaligned cache line split load to NC or Device memory, tagged with poison or external error on its first half, might cause data corruption

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

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

Configurations Affected
The erratum affects all configurations.

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

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

Workaround
There is no workaround for this erratum.
1051464
CTI trigger occurring on same cycle PREADYCD is received might cause CTI trigger to be missed

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
No workaround is required for this erratum.
1057923
Extra instruction might be executed during Halting Step when stepping WFI, WFE, and some self-synchronizing system register writes

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
There is no workaround for this erratum.
1069401
Debug APB accesses to the ELA RAM might return incorrect data

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

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

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

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

Configurations Affected
This erratum affects all configurations with ELA set to TRUE.

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

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

Workaround
This erratum can be avoided by inserting two writes to the ELA Lock Access Register (LAR) before writing the RRAR register.
1096402

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

If tracing only EL0, then no workaround is required.
Otherwise, setting TRCCONFIGR.RS = 0 to disable return stack is the workaround.
1109624
Continuous failing STREX with VA alias access outside mispredicted exclusive sequence (LDREX/STREX) loop might cause livelock

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

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

Configurations Affected
The erratum affects all configurations.

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

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

Workaround
This erratum is not expected to be encountered in real software. There is no workaround for this erratum.
1119735
16-bit T32 instruction close to breakpoint location might cause early breakpoint exception

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum. This situation can be detected by reading the contents of the appropriate ELR_ELx register after the breakpoint exception has been taken.
**1126105**

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

**Status**

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

**Description**

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

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

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

**Implications**

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

**Workaround**

The Index field for the RAMINDEX operation can be adjusted appropriately to access the desired L1 instruction cache data array entry.
1144394

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1192279
IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1194748
The ERXADDR_EL1 register might report an incorrect physical address for an L1 data tag RAM single-bit correctable ECC error

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
There is no workaround for this erratum.
1194749
ERR0MISC0 might report incorrect BANK and SUBBANK values for parity errors in L1 instruction cache data array

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
There is no workaround for this erratum.
1214504
Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by inserting a DSB after every instruction that accesses the L1 data TLB directly.
1227053
Streaming writes to memory mapped Non-shareable and write-back might cause data corruption because of reordering

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

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

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

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

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

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

Workaround
This erratum can be avoided by mapping all write-back memory as Inner or Outer Shareable.
1227629

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1244984
Illegal return event might corrupt PSTATE.UA0

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

Description
An illegal return event from AArch64 state erroneously updates PSTATE.UA0 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.UA0 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.UA0 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.UA0 has no impact on instruction execution until returning from the Illegal Execution state exception handler.

Workaround
No workaround is required for this erratum.
1256788

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

**Status**

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

**Description**

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

**Configurations Affected**

This erratum affects all configurations.

**Conditions**

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

**Implications**

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

**Workaround**

There is no workaround for this erratum.
1264383
Write-Back load after two Device-nG* stores to the same physical address might get invalid data

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1346756
TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0

Status
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r2p0, r3p0, r3p1, and r4p0. 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.
1349291
Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be mitigated by treating all SErrors reported with type Unrecoverable (UEU) as type Uncontainable (UC).
1356341
L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

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

Implications

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

Workaround

There is no workaround for this erratum.
1395332
Read from PMCCNTR in AArch32 might return corrupted data

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum is not expected to require a workaround.
1406411
MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.{N,C,Z,V,GE} on debug exit

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
The erratum can be avoided by setting CPUACTLR_EL1[45] to 1 prior to exiting from debug state. Power consumption in the core will be higher when CPUACTLR_EL1[45] is 1, as this prevents dynamic clock gating within sections of the core.
1408724

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

Status

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

Description

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

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

Configurations Affected

This erratum affects all configurations.

Conditions:

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

Implications

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

Workaround

No workaround is required. The programming error should be evident to users from the ETM trace information pertaining to the indirect branch with a malformed branch target address and trace information from its resultant Instruction Abort.
Ordering violation might occur when a load encounters an L1 tag RAM single bit ECC error when a snoop request targets the same line

Status

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

Description

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

Configurations Affected

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

Conditions

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

Implications

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

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

Workaround

There is no workaround for this erratum.
1430754

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions

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

Implications

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

Workaround

Any interrupt will break the core out of the deadlock state.
1487185
Waypoints from previous session might cause single-shot comparator match when trace enabled

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

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.
1490853
TRCIDR3.CCITMIN value is incorrect

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

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.
1514034

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

Status

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

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.
1523502
CPUECTLR_EL1 controls for the MMU have no affect

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround.
1627784
ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect

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

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

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

Workaround
There is no workaround for this erratum.
1655746
MRC read of DBGDSCRint into APSR_nzcv might produce wrong results and lead to corruption

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

Description
In AArch32, MRC reads of DBGDSCRint into destination APSR_nzcv (Rt=15) always produce a result of 0. Also, if there is a younger MRC or MRRC read to any accessible register following the DBGDSCRint read into APSR_nzcv, then the younger read result might be corrupted.

Configurations Affected
This erratum affects all configurations.

Conditions
1. The core is in AArch32 state at EL0.
2. An MRC read of DBGDSCRint into APSR_nzcv (Rt=15) occurs.

Implications
If the above conditions are met, then:

1. APSR_nzcv is always written with 0.
2. Under specific microarchitectural timing conditions in AArch32 EL0, a subsequent MRC or MRRC might be corrupted.

Workaround
Directly read DBGDSCRint with an MRC instruction into a general-purpose register (R0-R14), and then write that general-purpose register to the flags by doing an MSR APSR_f. To avoid the possible corruption, add an ISB instruction before any subsequent MRC or MRRC instructions.
1662732

Cache maintenance performed on an instruction being actively modified by another PE might cause unexpected behavior

Status

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

Description

A PE that performs a data cache clean and invalidate instruction (DC CIVAC) to memory locations that contain code which is being actively modified by another PE might not execute the newly written code, which is required by Arm architecture.

Configurations Affected

This erratum affects configurations with instruction cache coherency enabled.

Conditions

1. PE0 writes new instructions to memory location A, and sets a flag indicating that the new code is ready:
   1. Store A, <new code>
   2. DMB
   3. store flag B
2. PE1 executes a DC CIVAC instruction to location A
3. PE1 executes:
   1. LOOP: load flag B
   2. CBZ LOOP
   3. ISB
   4. branch A

Implications

If the above conditions are met, PE1 might not execute the code that is written by PE0, which required by the Arm architecture. Arm does not expect that the code sequence that is executed by PE1 appears in normal code.

Workaround

If the DC CIVAC executed by PE1 is necessary, follow it with a DMB instruction.
1694299

Instruction sampling bias exists in SPE implementation

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

1. SPE configured and utilized on PE.

Implications

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

Workaround

There is no workaround.
1697035

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

Status

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

Conditions

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

Implications

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

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

Workaround

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.