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 © 2019 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.

Web address

http://www.arm.com/.
Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on this document

If you have comments on content then send an e-mail to errata@arm.com giving:

- The document title.
- The document number: SDEN-1152370.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.
## Contents

<p>| INTRODUCTION | 7 |
| ERRATA SUMMARY TABLE | 13 |
| 1316063 | Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-write ordering violation | 16 |
| 1160841 | Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock | 17 |
| 1177367 | Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation Status | 18 |
| 1191167 | MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result | 19 |
| 1204882 | The exclusive monitor might end up tracking an incorrect cache line in the presence of a VA-alias, causing a false pass on the exclusive access sequence | 20 |
| 1220737 | Streaming store under specific conditions might cause deadlock or data corruption | 21 |
| 1253791 | Multiple floating-point divides/square roots concurrently completing back-to-back and flushing back-to-back might cause data corruption Status | 22 |
| 1262841 | Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation | 23 |
| 1273521 | A T32 instruction inside an IT block followed by mispredicted speculative instruction stream might cause a deadlock | 24 |
| 1450698 | Software Step might prevent interrupt recognition | 25 |
| 1467687 | Branch prediction for an ERET cached in the instruction cache might cause a deadlock | 27 |
| 1508412 | NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock | 28 |
| 1515815 | The core might execute multiple instructions before taking a software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache | 29 |
| 1286809 | Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation | 29 |
| 1418842 | MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data | 31 |
| 1542418 | The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches | 32 |
| 1148171 | ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array | 32 |
| 1151664 | Direct access to internal memory for L2 TLB might not update IDATAAn_EL3 registers | 34 |
| 1162083 | 16-bit T32 instruction close to breakpoint location may cause early breakpoint exception | 35 |
| 1185469 | Exception packet for return stack match might return incorrect [E1:E0] field | 36 |
| 1192280 | IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level | 37 |
| 1207839 | Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error | 38 |
| 1220404 | Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry | 39 |
| 1220843 | ERROSTATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported | 40 |
| 1244986 | Illegal return event might corrupt PSTATE.UA0 | 41 |
| 1256789 | Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error | 42 |</p>
<table>
<thead>
<tr>
<th>Issue Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1262908</td>
<td>Write-Back load after two Device-nG* stores to the same physical address might get invalid data</td>
</tr>
<tr>
<td>1328683</td>
<td>Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError</td>
</tr>
<tr>
<td>1346768</td>
<td>TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0</td>
</tr>
<tr>
<td>1355135</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>1395535</td>
<td>Read from PMCCNTR in AArch32 might return corrupted data</td>
</tr>
<tr>
<td>1405548</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>1415321</td>
<td>LDREX-STREX might succeed incorrectly when an intervening store occurs and LDREX detects a single-bit ECC error on the cache line in the L1 data cache tag RAM</td>
</tr>
<tr>
<td>1421023</td>
<td>Portions of the branch target address recorded in ETM trace information are incorrect for an indirect branch with a malformed branch target address</td>
</tr>
<tr>
<td>1487187</td>
<td>Waypoints from previous session might cause single-shot comparator match when trace enabled</td>
</tr>
<tr>
<td>1488613</td>
<td>An unaligned load might initiate a prefetch request which crosses a page boundary</td>
</tr>
<tr>
<td>1491015</td>
<td>TRCIDR3.CCITMIN value is incorrect</td>
</tr>
<tr>
<td>1514033</td>
<td>Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE</td>
</tr>
<tr>
<td>1519163</td>
<td>AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle</td>
</tr>
<tr>
<td>1522097</td>
<td>The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error</td>
</tr>
<tr>
<td>1523503</td>
<td>CPUECTR_EL1 controls for the MMU have no affect</td>
</tr>
<tr>
<td>1610369</td>
<td>ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect</td>
</tr>
<tr>
<td>1624431</td>
<td>CPUAMEVTYPER4_EL0 register cannot be written</td>
</tr>
</tbody>
</table>
r0p0 implementation fixes

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

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

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

r1p0 implementation fixes

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

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

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

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[3] | 1450698 Software Step might prevent interrupt recognition |

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

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

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

**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 13 identifies errata that have been fixed in each product revision.

### 08-Nov-2019: 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>1467687</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>Branch prediction for an ERET cached in the instruction cache might cause a deadlock</td>
</tr>
<tr>
<td>1508412</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatB</td>
<td>NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock</td>
</tr>
<tr>
<td>1542418</td>
<td>New</td>
<td>Programmer</td>
<td>CatB (rare)</td>
<td>The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches</td>
</tr>
<tr>
<td>1514033</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>1519163</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle</td>
</tr>
<tr>
<td>1522097</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error</td>
</tr>
<tr>
<td>1523503</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>CPUECTLR_EL1 controls for the MMU have no affect</td>
</tr>
<tr>
<td>1610369</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>1624431</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>CPUAMEVTYPER4_EL0 register cannot be written</td>
</tr>
</tbody>
</table>

### 15-Jul-2019: 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>1508412</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock</td>
</tr>
<tr>
<td>1515815</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>The core might execute multiple instructions before taking a software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache</td>
</tr>
<tr>
<td>1487187</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>1488613</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>An unaligned load might initiate a prefetch request which crosses a page boundary</td>
</tr>
<tr>
<td>1491015</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>TRCIDR3.CCITMIN value is incorrect</td>
</tr>
</tbody>
</table>
## 24-May-2019: 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>1450698</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Software Step might prevent interrupt recognition</td>
</tr>
<tr>
<td>1467687</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Branch prediction for an ERET cached in the instruction cache might cause a deadlock</td>
</tr>
</tbody>
</table>

## 29-Mar-2019: 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>1418842</td>
<td>New</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>1328683</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>1346768</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>1355135</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>1395535</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Read from PMCCNTR in AArch32 might return corrupted data</td>
</tr>
<tr>
<td>1405548</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>1415321</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>LDREX-STREX might succeed incorrectly when an intervening store occurs and LDREX detects a single-bit ECC error on the cache line in the L1 data cache tag RAM</td>
</tr>
<tr>
<td>1421023</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Portions of the branch target address recorded in ETM trace information are incorrect for an indirect branch with a malformed branch target address</td>
</tr>
</tbody>
</table>

## 15-Mar-2019: 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></td>
<td></td>
<td></td>
<td></td>
<td>No new or updated errata in this document version.</td>
</tr>
</tbody>
</table>

## 03-Dec-2018: Changes in document version 4.0

<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>1316063</td>
<td>New</td>
<td>Programmer</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>
</tr>
</tbody>
</table>
### 05-Oct-2018: 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>1160841</td>
<td>Updated</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>1177367</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>1191167</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>1204882</td>
<td>Updated</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>1220737</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>Streaming store under specific conditions might cause deadlock or data corruption</td>
</tr>
<tr>
<td>1253791</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>1262841</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>1273521</td>
<td>New</td>
<td>Programmer</td>
<td>CatB</td>
<td>A T32 instruction inside an IT block followed by mispredicted speculative instruction stream might cause a deadlock</td>
</tr>
<tr>
<td>1148171</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array</td>
</tr>
<tr>
<td>1151664</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>Direct access to internal memory for L2 TLB might not update IDATA_EL3 registers</td>
</tr>
<tr>
<td>1162083</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>16-bit T32 instruction close to breakpoint location may cause early breakpoint exception</td>
</tr>
<tr>
<td>1185469</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>1192280</td>
<td>Updated</td>
<td>Programmer</td>
<td>CatC</td>
<td>IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level</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>1207839</td>
<td>Updated</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>1220404</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry</td>
</tr>
<tr>
<td>1220843</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>ERRSTATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported</td>
</tr>
<tr>
<td>1244986</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Illegal return event might corrupt PSTATE.UA0</td>
</tr>
<tr>
<td>1256789</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>1262908</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>

**27-Jul-2018: 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>1160841</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>1177367</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 Status</td>
</tr>
<tr>
<td>1191167</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>1204882</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>1162083</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>16-bit T32 instruction close to breakpoint location may cause early breakpoint exception</td>
</tr>
<tr>
<td>1185469</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>1192280</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level</td>
</tr>
<tr>
<td>1207839</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>
</tbody>
</table>

**16-May-2018: 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></td>
</tr>
<tr>
<td>Issue ID</td>
<td>Status</td>
<td>Role</td>
<td>Category</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------</td>
<td>----------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1148171</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array</td>
</tr>
<tr>
<td>1151664</td>
<td>New</td>
<td>Programmer</td>
<td>CatC</td>
<td>Direct access to internal memory for L2 TLB might not update IDATan_EL3 registers</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>1316063</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>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1160841</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</td>
<td>r1p0</td>
</tr>
<tr>
<td>1177367</td>
<td>CatB</td>
<td>Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation. Status</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1191167</td>
<td>CatB</td>
<td>MRC read following MRRC read of specific Generic Timer in AArch32 might give incorrect result</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1204882</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</td>
<td>r1p0</td>
</tr>
<tr>
<td>1220737</td>
<td>CatB</td>
<td>Streaming store under specific conditions might cause deadlock or data corruption</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1253791</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. Status</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1262841</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</td>
<td>r1p0</td>
</tr>
<tr>
<td>1273521</td>
<td>CatB</td>
<td>A T32 instruction inside an IT block followed by mispredicted speculative instruction stream might cause a deadlock</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1450698</td>
<td>CatB</td>
<td>Software Step might prevent interrupt recognition</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1467687</td>
<td>CatB</td>
<td>Branch prediction for an ERET cached in the instruction cache might cause a deadlock</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1508412</td>
<td>CatB</td>
<td>NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1515815</td>
<td>CatB</td>
<td>The core might execute multiple instructions before taking a software step exception or halt step exception when the executing</td>
<td>r0p0, r1p0</td>
<td>r1p1</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>1286809</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</td>
<td>r1p1</td>
</tr>
<tr>
<td>1418842</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</td>
<td>r1p1</td>
</tr>
<tr>
<td>1542418</td>
<td>CatB (rare)</td>
<td>The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1148171</td>
<td>CatC</td>
<td>ERR0MISC0 might report incorrect BANK and SUBBANK values for transient parity errors in L1 instruction cache data array</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1151664</td>
<td>CatC</td>
<td>Direct access to internal memory for L2 TLB might not update IDATAn_EL3 registers</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1162083</td>
<td>CatC</td>
<td>16-bit T32 instruction close to breakpoint location may cause early breakpoint exception</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1185469</td>
<td>CatC</td>
<td>Exception packet for return stack match might return incorrect [E1:E0] field</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1192280</td>
<td>CatC</td>
<td>IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1207839</td>
<td>CatC</td>
<td>Software step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1220404</td>
<td>CatC</td>
<td>Direct access to L1 data TLB might report incorrect value of valid bit of the corresponding TLB entry</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1220843</td>
<td>CatC</td>
<td>ERRORSTATUS.SERR encoding is incorrect for error responses from slave and deferred data errors from slave which are not supported</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1244986</td>
<td>CatC</td>
<td>Illegal return event might corrupt PSTATE.UA0</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1256789</td>
<td>CatC</td>
<td>Halting step might see extra instruction executed for some loads when crossed with snoop invalidation or ECC error</td>
<td>r0p0</td>
<td>r1p0</td>
</tr>
<tr>
<td>1262908</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</td>
<td>r1p0</td>
</tr>
<tr>
<td>1328683</td>
<td>CatC</td>
<td>Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1346768</td>
<td>CatC</td>
<td>TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0</td>
<td>r0p0, r1p0, r1p1</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>1355135</td>
<td>CatC</td>
<td>LTD_CACHE access related PMU events and L1D_TLB access related PMU events increments on instructions/micro-operations excluded from these events</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1395535</td>
<td>CatC</td>
<td>Read from PMCCNTR in AArch32 might return corrupted data</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1405548</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</td>
<td>r1p1</td>
</tr>
<tr>
<td>1415321</td>
<td>CatC</td>
<td>LDREX-STREX might succeed incorrectly when an intervening store occurs and LDREX detects a single-bit ECC error on the cache line in the L1 data cache tag RAM</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1421023</td>
<td>CatC</td>
<td>Portions of the branch target address recorded in ETM trace information are incorrect for an indirect branch with a malformed branch target address</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1487187</td>
<td>CatC</td>
<td>Waypoints from previous session might cause single-shot comparator match when trace enabled</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1488613</td>
<td>CatC</td>
<td>An unaligned load might initiate a prefetch request which crosses a page boundary</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1491015</td>
<td>CatC</td>
<td>TRCIDR3.CCITMIN value is incorrect</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1514033</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</td>
<td>r1p1</td>
</tr>
<tr>
<td>1519163</td>
<td>CatC</td>
<td>AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1522097</td>
<td>CatC</td>
<td>The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1523503</td>
<td>CatC</td>
<td>CPUECTR_EL1 controls for the MMU have no affect</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
<tr>
<td>1610369</td>
<td>CatC</td>
<td>ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect</td>
<td>r0p0, r1p0, r1p1</td>
<td>Open</td>
</tr>
<tr>
<td>1624431</td>
<td>CatC</td>
<td>CPUAMEVTYPER4_EL0 register cannot be written</td>
<td>r0p0, r1p0</td>
<td>r1p1</td>
</tr>
</tbody>
</table>
Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

1316063

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

Status

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

Description

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

Configurations Affected

The erratum affects all multi-core configurations.

Conditions.

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

Implications

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

Workaround

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

1160841
Continuous failing STREX because of another core snooping from speculatively executed atomic behind constantly mispredicted branch might cause livelock

Status

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

Description

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

Configurations Affected

The erratum affects all configurations.

Conditions

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

Implications

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

Workaround

Set CPUACTLR2_EL1[0] to 1 and CPUACTLR2_EL1[15] to 1.
1177367

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

When context-switching the register state for an out-of-context translation regime, system software at EL2 or above must ensure that all intermediate states during the context-switch would report a level 0 translation fault in response to an AT instruction targeting the out-of-context translation regime. Note that a workaround is only required if the system software contains an AT instruction as part of an executable page.
1191167

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

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workarounds
The erratum can be avoided by trapping MRC/MCR/MRRC/MCRR accesses in AArch32 to the affected registers and doing the equivalent code sequence in the trap handler. To trap the CNT* accesses, set CNTKCTL_EL1.[EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN] to 0. If HCR_EL2.[E2H,TGE]=[1,1] then set CNTHCTL_EL2.[EL0PTEN, EL0VTEN, EL0VCTEN, EL0PCTEN] to 0. The following registers will be trapped: CNTP_CTL, CNTP_CVAL, CNTP_TVAL, CNTV_CTL, CNTV_CVAL, CNTV_TVAL, CNTPCT, CNTVCT, CNTFREQ.
1204882

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided if software sets CPUACTLR2_EL1 bit[11] to 0b1.
1220737
Streaming store under specific conditions might cause deadlock or data corruption

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

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

Configurations Affected
The erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUECTLR_EL1[25:24] to 0b11, which disables write streaming to the L2. This will have an impact on performance for streaming workloads.
1253791

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by setting CPUACTLR3_EL1[10] to 1, which prevents parallel execution of divide and square root instructions.
1262841
Translation access hitting a prefetched L2 TLB entry under specific conditions might corrupt the L2 TLB leading to an incorrect translation

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

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

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUECTRL_EL1[51] to 1, which disables the MMU hardware prefetcher. Setting this bit might have a small impact on performance.
1273521
A T32 instruction inside an IT block followed by a mispredicted speculative instruction stream might cause a deadlock

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUACTLR_EL1[13] to 1 to increase the mispredict to fetch latency, which will have some impact on performance.
1450698
Software Step might prevent interrupt recognition

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

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

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

If a workaround is required when MDSCR_EL1.KDE is set to 1, then please contact Arm.
1467687
Branch prediction for an ERET cached in the instruction cache might cause a deadlock

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

```
LDR x0,=0x3
MSR S3_6_c15_c8_0,x0
LDR x0,=0xF3D08000
MSR S3_6_c15_c8_2,x0
LDR x0,=0xFFF0F0FF
MSR S3_6_c15_c8_3,x0
LDR x0,=0x80000002003FF
MSR S3_6_c15_c8_1,x0
ISB
```
1508412
NC/Device Load and Store Exclusive or PAR-Read collision can cause deadlock

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by having the hardware insert a DMB SY before and after accessing PAR_EL1 and inserting a DMB SY before and after a store-exclusive instruction. This can be done through the following write sequence to several IMPLEMENTATION DEFINED registers. The code sequence should be applied early in the boot sequence prior to any of the possible errata conditions being met.

```
LDR x0,=0x0
MSR S3_6_c15_c8_0,x0
LDR x0,= 0xEE070F14
MSR S3_6_c15_c8_2,x0
LDR x0,= 0xFFFF0FFF
MSR S3_6_c15_c8_3,x0
LDR x0,=0x4005003FF
MSR S3_6_c15_c8_1,x0

LDR x0,=0x1
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8400000
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00fff00000
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4005007FF
MSR S3_6_c15_c8_1,x0

LDR x0,=0x2
MSR S3_6_c15_c8_0,x0
LDR x0,=0x00e8c00040
MSR S3_6_c15_c8_2,x0
LDR x0,=0x00fff00040
MSR S3_6_c15_c8_3,x0
LDR x0,= 0x4005007FF
MSR S3_6_c15_c8_1,x0
ISB
```
1515815
The core might execute multiple instructions before taking a software step exception or halt step exception when the executing instruction resides in the L0 Macro-op cache

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

Category B (rare)
1286809
Modification of the translation table for a virtual page which is being accessed by an active process might lead to read-after-read ordering violation

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

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

Configurations Affected
The erratum affects all multi-core configurations.

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

Implications

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

Workaround

This erratum can be avoided by executing the TLB invalidate and DSB instructions a second time before modifying the translation table of a virtual page that is being accessed by an active process.
1418842
MRRC reads of some Generic Timer system registers in AArch32 mode might return corrupt data

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

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

- CNTP_CTL.
- CNTP_CVAL.
- CNTP_TVAL.
- CNTV_CTL.
- CNTV_CVAL.
- CNTV_TVAL.
- CNTPCT.
- CNTVCT.
- CNTFRQ.
The core might fetch a stale instruction from the L0 Macro-op cache which violates the ordering of instruction fetches

Status

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

Description

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

Configurations Affected

This erratum affects all multi-core configurations.

Conditions

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

Implications

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

Workaround

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

Category C

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

Status

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

Description

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

Configurations affected

This erratum affects all configurations with CORE_CACHE_PROTECTION set to TRUE.
Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1151664
Direct access to internal memory for L2 TLB might not update IDATAn_EL3 registers

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be avoided by setting CPUACTLR_EL1[14] to 1 for performing direct access to internal memory.
1162083
16-bit T32 instruction close to breakpoint location may cause early breakpoint exception

Status

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

Description

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

Configurations affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum. This situation can be detected by reading the contents of the appropriate ELR_ELx register after the breakpoint exception has been taken.
1185469
Exception packet for return stack match might return incorrect [E1:E0] field

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

If tracing only EL0, then no workaround is required.
Otherwise, setting TRCCONFIGR.RS = 0 to disable return stack is the workaround.
1192280
IMPLEMENTATION DEFINED fault for unsupported atomic operations is not routed to proper Exception level

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

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

Configurations Affected
The erratum affects all configurations.

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

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

Workaround
There is no workaround.
1207839

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by inserting a DSB after every instruction that accesses the L1 data TLB directly.
1220843

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

- The core issues a copyback transaction (WriteBackFull, WriteEvictFull, Evict, or WriteNoSnpFull) which then receives an error response.
- The core receives data containing an error (Poison or DErr response), but the core caches cannot defer the error by marking the data as poisoned in its caches. This occurs when the core is configured with CORE_CACHE_PROTECTION set to FALSE, or when ERR0CTRL.ED is 0.

Implications

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

Workarounds

There is no workaround for this erratum.
1244986

Illegal return event might corrupt PSTATE.UA0

Status

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

Description

An illegal return event from AArch64 state erroneously updates PSTATE.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.
1256789

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

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1262908
Write-Back load after two Device-nG* stores to the same physical address might get invalid data

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

There is no workaround for this erratum.
1328683
Uncontainable (UC) SError might be incorrectly logged as an Unrecoverable (UEU) SError

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum can be mitigated by treating all SErrors reported with type Unrecoverable (UEU) as type Uncontainable (UC).
1346768
TLBI does not treat upper ASID bits as zero when TCR_EL1.AS is 0

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

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

Configurations Affected
The erratum affects all configurations.

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

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

Workaround
This erratum can be avoided if software is properly writing zero to RES0 bits.
1355135
L1D_CACHE access related PMU events and L1D_TLB access related PMU events increment on instructions/micro-operations excluded from these events

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

Workaround
There is no workaround for this erratum.
1395535
Read from PMCCNTR in AArch32 might return corrupted data

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
This erratum is not expected to require a workaround.
1405548
MSR DSPSR_EL0 while in debug state might not correctly update PSTATE.(N,C,Z,V,GE) on debug exit

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
The erratum can be avoided by setting CPUACTLR_EL1[45] to 1 prior to exiting from debug state. Power consumption in the core will be higher when CPUACTLR_EL1[45] is 1, as this prevents dynamic clock gating within sections of the core.
1415321

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

Status

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

Description

If a core:

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

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

Configurations Affected

The erratum affects all multicore configurations with CORE_CACHE PROTECTION= 1.

Conditions

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

Implications

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

Workaround

This erratum can be avoided by setting CPUACTLR3_EL1[57].
1421023

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

Status

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

Description

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

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

Configurations Affected

This erratum affects all configurations.

Conditions:

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

Implications

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

Workaround

No workaround is required. The programming error should be evident to users from the ETM trace information pertaining to the resultant Instruction Abort.
1487187
Waypoints from previous session might cause single-shot comparator match when trace enabled

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

Between tracing sessions, set the core ETM to enter a prohibited region either instead of or in addition to disabling the ETM.
1488613
An unaligned load might initiate a prefetch request which crosses a page boundary

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
Arm does not expect read-sensitive Device memory to be mapped to a physical address which overlaps with a 64KB aligned physical address region belonging to Normal memory, therefore no workaround is necessary.
1491015
TRCIDR3.CCITMIN value is incorrect

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
The value for the TRCIDR3.CCITMIN field should be treated as 0x4.
1514033
Error Synchronization Barrier (ESB) instruction execution with a pending masked Virtual SError might not clear HCR_EL2.VSE

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

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

Configurations Affected
This erratum affects all configurations.

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

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

Workaround
A workaround is not expected to be required. This is because existing software only executes ESB instructions at EL2 and above. If your software executes ESB instructions at EL1 with the conditions described above, then contact Arm support for more details.
1519163
AMU Counter INST_RETIRED does not increment correctly when 16 instructions retire in same cycle

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

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

Configurations Affected
This erratum affects all configurations.

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

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

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

Workaround
To workaround this issue, NOP elimination must be disabled. This is done by setting both CPUACTLR_EL1[28] and CPUACTLR_EL1[26] to 1'b1. The performance delta of this workaround is expected to be small.
1522097
The core might detect a breakpoint exception one instruction earlier than the programmed location when the L0 Macro-op cache contains an instruction that is affected by a parity error

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

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

Configurations Affected
This erratum affects the configuration with CORE_CACHE_PROTECTION = 1.

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

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

Workaround
This erratum has no workaround.
1523503

CPUECTRL_EL1 controls for the MMU have no affect

Status

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

Description

The CPUECTRL_EL1 register contains IMPLEMENTATION DEFINED configuration and control options for the MMU. The MMU bits affected by this erratum are CPUECTRL_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 CPUECTRL_EL1[54:46] from reset values have no affect.

Implications

Software attempts to change the functionality or performance of the core by changing reset values of CPUECTRL_EL1[54:46] have no affect. The value is updated in the register correctly, such that any subsequent read of the CPUECTRL_EL1 register returns the expected data, however, the modifications have no affect on the behavior of the core.

Workaround

There is no workaround.
1610369
ERR0MISC0_EL1.SUBARRAY value for ECC errors in the L1 data cache might be incorrect

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

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

Configurations Affected
The erratum affects configurations with CORE_CACHE_PROTECTION set to TRUE.

Conditions
1. Load, store or atomic instructions access multiple banks of the L1 data cache.
2. One of the banks accessed has an ECC error.

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

Workaround
This erratum has no workaround.
1624431
CPUAMEVTYPER4_EL0 register cannot be written

Status

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

Description

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

Configurations Affected

This erratum affects all configurations.

Conditions

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

Implications

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

Workaround

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