



## Arm DynamIQ Shared Unit-110 MP108

### Software Developer Errata Notice

Date of issue: 30-Mar-2022

Non-Confidential

Document version: v11.0

Copyright © 2022 Arm® Limited (or its affiliates). All rights reserved.

Document ID: SDEN-1781796

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, third party 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 <https://www.arm.com/company/policies/trademarks>.

Copyright © 2022 Arm® Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

## Confidentiality status

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

Unrestricted Access is an Arm internal classification.

## Product status

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

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm DynamIQ Shared Unit-110 MP108, create a ticket on <https://support.developer.arm.com>.

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

## Inclusive language commitment

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

If you find offensive language in this document, please email [terms@arm.com](mailto:terms@arm.com).

# Contents

<b>Introduction</b>	<b>8</b>
Scope	8
Categorization of errata	8
<b>Change Control</b>	<b>9</b>
<b>Errata summary table</b>	<b>12</b>
<b>Errata descriptions</b>	<b>15</b>
Category A	15
2125021 Memory traffic might cause a deadlock	15
2090737 Heavy snoop traffic might cause a deadlock	16
1873142 Loss of coherency after exclusive load/store	17
Category A (rare)	18
Category B	19
2381090 TLB not invalidated in complex power transitions	19
2415004 DSU might lose coherency after data cache clean	21
2426404 DVM might violate CHI protocol during cluster powerdown	23
2313941 SCLK clock gating might prevent cluster register access	25
2309643 CLUSTERL3HIT and CLUSTERL3MISS registers incorrectly decrement	27
2304881 Power transition from all slices to one slice might cause deadlock	28
2282666 CLUSTERL3HIT and CLUSTERL3MISS registers count incorrectly	29
2091675 Cache debug access while in FUNC_RET might lead to deadlock	30
2089240 Large number of outstanding ACP read transactions might deadlock	31
2085140 Debug accesses during core power off might deadlock	32
2057044 Debug accesses when core powered off might deadlock	34
2055944 FUNC_RET while all cores powered off might deadlock	35
1976458 Cluster PMU or ELA triggers affect core debug	37
1970782 SCLK clock gating might cause a deadlock	39
1935474 Exclusive sequence to 64 bit Peripheral Port does not succeed	40
1911265 Complex VPU powerdown might deadlock	41
1905020 Interrupt outputs remain asserted after a core is powered off	42
1887098 Isolation incorrectly applied for OFF_EMU power modes	43
1869818 ATB clock gating might cause deadlock	44
1814712 APB writes can incorrectly update some cluster PMU registers	45
Category B (rare)	46
1827720 PPU can produce spurious interrupt if written to during initialization sequence	46

Category C	49
2457882 CLUSTERPMSSRR bits cannot be cleared	49
2375351 L3 tag RAM chip enable pins incorrectly asserted in RAM power transitions	50
2330050 SCLK clock gating might prevent cluster register access when PERIPHCLK at a higher frequency	51
2349173 MPAMCFG_CPBMS.S_EXCL bit not functioning	53
2301655 Cache debug access might return incorrect data	54
2301453 Incorrect CLUSTERPMU_PMCEID0/IMP_CLUSTERPMCEID0_EL1 value	55
2261337 PPT_ACCESS access PMU event counts incorrectly for 64-bit AXI Peripheral Port	56
2165755 Debug APB old CTI address space accessible through new address map	57
2135307 DSU might lose coherency after snoop filter error	58
2121479 Debug APB accesses to reserved core address is possible when the core is powered off	59
2120242 ACP atomic transaction might incorrectly report error	60
2049666 L3 cache allocate PMU event counts incorrectly	62
1985295 Cluster PMU overflow flag might get set when counters are disabled	63
1971432 L3 cache access PMU events count incorrectly	64
1965174 Utility bus accesses to reserved address range might affect other accesses	66
1961495 Cache debug access while transactions are ongoing might lead to deadlock	67
1941771 DBG_RECOV transition when cluster is OFF might deadlock	68
1906346 PMU events and RAS errors not counted correctly in some 2 node configurations	70
1892065 Interconnect DErr on dirty data not reported in RAS registers	71
1877057 Cluster transition to OFF_EMU while core powers on might deadlock	72
1861790 AXI master PMU events count incorrectly	73
1854639 Incorrect RAS CIDR1 value	75
1852252 Incorrect PRIDR0 value	76
1836065 OFF_EMU to OFF power transitions will deadlock	77
1799687 Interconnect errors not reported in RAS registers	78
1795881 Trace flush may not flush all data	80
1791722 ECC errors in L3 Tag RAMs can cause data corruption and loss of coherency	81

## r3p1 implementation fixes

Note the following errata might be fixed in some implementations of r3p1. This can be determined by reading the IMP\_CLUSTERREVIDR\_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR[0]	2282666	CLUSTERL3HIT and CLUSTERL3MISS registers count incorrectly
REVIDR[2]	2309643	CLUSTERL3HIT and CLUSTERL3MISS registers incorrectly decrement
REVIDR[3]	2304881	Power transition from all slices to one slice might cause deadlock
REVIDR[5]	2313941	SCLK clock gating might prevent cluster register access

Note that there is no change to the IMP\_CLUSTERIDR\_EL1 which remains at r3p1 but the IMP\_CLUSTERREVIDR\_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of IMP\_CLUSTERIDR\_EL1 and IMP\_CLUSTERREVIDR\_EL1.

## r3p0 implementation fixes

Note the following errata might be fixed in some implementations of r3p0. This can be determined by reading the IMP\_CLUSTERREVIDR\_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR[0]	2282666	CLUSTERL3HIT and CLUSTERL3MISS registers count incorrectly
-----------	---------	--

Note that there is no change to the IMP\_CLUSTERIDR\_EL1 which remains at r3p0 but the IMP\_CLUSTERREVIDR\_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of IMP\_CLUSTERIDR\_EL1 and IMP\_CLUSTERREVIDR\_EL1.

## r2p1 implementation fixes

Note the following errata might be fixed in some implementations of r2p1. This can be determined by reading the IMP\_CLUSTERREVIDR\_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR[0]	2085140	Debug accesses during core power off might deadlock
REVIDR[1]	2090737	Heavy snoop traffic might cause a deadlock
REVIDR[3]	2125021	Memory traffic might cause a deadlock

Note that there is no change to the IMP\_CLUSTERIDR\_EL1 which remains at r2p1 but the IMP\_CLUSTERREVIDR\_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of IMP\_CLUSTERIDR\_EL1 and IMP\_CLUSTERREVIDR\_EL1.

## r1p0 implementation fixes

Note the following errata might be fixed in some implementations of r1p0. This can be determined by reading the IMP\_CLUSTERREVIDR\_EL1 register where a set bit indicates that the erratum is fixed in this part.

REVIDR[6]	1887098	Isolation incorrectly applied for OFF_EMU power modes
-----------	---------	---

Note that there is no change to the IMP\_CLUSTERIDR\_EL1 which remains at r1p0 but the IMP\_CLUSTERREVIDR\_EL1 is updated to indicate which errata are corrected. Software will identify this release through the combination of IMP\_CLUSTERIDR\_EL1 and IMP\_CLUSTERREVIDR\_EL1.

# Introduction

## Scope

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

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

## Categorization of errata

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

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



# Change Control

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

## 30-Mar-2022: Changes in document version v11.0

ID	Status	Area	Category	Summary
<a href="#">2426404</a>	New	Programmer	Category B	DVM might violate CHI protocol during cluster powerdown
<a href="#">2415004</a>	New	Programmer	Category B	DSU might lose coherency after data cache clean
<a href="#">2381090</a>	New	Programmer	Category B	TLB not invalidated in complex power transitions
<a href="#">2301453</a>	New	Programmer	Category C	Incorrect CLUSTERPMU_PMCEID0/IMP_CLUSTERPMCEID0_EL1 value
<a href="#">2301655</a>	New	Programmer	Category C	Cache debug access might return incorrect data
<a href="#">2349173</a>	New	Programmer	Category C	MPAMCFG_CPBMS_S_EXCL bit not functioning
<a href="#">2330050</a>	New	Programmer	Category C	SCLK clock gating might prevent cluster register access when PERIPHCLK at a higher frequency
<a href="#">2375351</a>	New	Programmer	Category C	L3 tag RAM chip enable pins incorrectly asserted in RAM power transitions
<a href="#">2457882</a>	New	Programmer	Category C	CLUSTERPMSSRR bits cannot be cleared

## 08-Oct-2021: Changes in document version v10.0

ID	Status	Area	Category	Summary
<a href="#">2304881</a>	New	Programmer	Category B	Power transition from all slices to one slice might cause deadlock
<a href="#">2309643</a>	New	Programmer	Category B	CLUSTERL3HIT and CLUSTERL3MISS registers incorrectly decrement
<a href="#">2313941</a>	New	Programmer	Category B	SCLK clock gating might prevent cluster register access

## 15-Sep-2021: Changes in document version v9.0

ID	Status	Area	Category	Summary
<a href="#">2282666</a>	New	Programmer	Category B	CLUSTERL3HIT and CLUSTERL3MISS registers count incorrectly

## 27-Jul-2021: Changes in document version v8.0

ID	Status	Area	Category	Summary
<a href="#">2085140</a>	Updated	Programmer	Category B	Debug accesses during core power off might deadlock
<a href="#">2120242</a>	New	Programmer	Category C	ACP atomic transaction might incorrectly report error
<a href="#">2121479</a>	New	Programmer	Category C	Debug APB accesses to reserved core address is possible when the core is powered off
<a href="#">2135307</a>	New	Programmer	Category C	DSU might lose coherency after snoop filter error
<a href="#">2165755</a>	New	Programmer	Category C	Debug APB old CTI address space accessible through new address map
<a href="#">2261337</a>	New	Programmer	Category C	PPT_ACCESS access PMU event counts incorrectly for 64-bit AXI Peripheral Port

**31-Mar-2021: Changes in document version v7.0**

ID	Status	Area	Category	Summary
<a href="#">2125021</a>	New	Programmer	Category A	Memory traffic might cause a deadlock

**05-Mar-2021: Changes in document version v6.0**

ID	Status	Area	Category	Summary
<a href="#">2090737</a>	New	Programmer	Category A	Heavy snoop traffic might cause a deadlock
<a href="#">2055944</a>	New	Programmer	Category B	FUNC_RET while all cores powered off might deadlock
<a href="#">2057044</a>	New	Programmer	Category B	Debug accesses when core powered off might deadlock
<a href="#">2085140</a>	New	Programmer	Category B	Debug accesses during core power off might deadlock
<a href="#">2089240</a>	New	Programmer	Category B	Large number of outstanding ACP read transactions might deadlock
<a href="#">2091675</a>	New	Programmer	Category B	Cache debug access while in FUNC_RET might lead to deadlock
<a href="#">2049666</a>	New	Programmer	Category C	L3 cache allocate PMU event counts incorrectly

**03-Nov-2020: Changes in document version v5.0**

ID	Status	Area	Category	Summary
<a href="#">1970782</a>	New	Programmer	Category B	SCLK gating might cause a deadlock
<a href="#">1976458</a>	New	Programmer	Category B	Cluster PMU or ELA triggers affect core debug
<a href="#">1961495</a>	New	Programmer	Category C	Cache debug access while transactions ongoing might lead to deadlock
<a href="#">1965174</a>	New	Programmer	Category C	Utility bus accesses to reserved address range might affect other accesses
<a href="#">1971432</a>	New	Programmer	Category C	L3 cache access PMU events count incorrectly
<a href="#">1985295</a>	New	Programmer	Category C	Cluster PMU overflow flag might get set when counters are disabled

**18-Sep-2020: Changes in document version v4.0**

ID	Status	Area	Category	Summary
<a href="#">1887098</a>	New	Programmer	Category B	Isolation incorrectly applied for OFF_EMU power modes
<a href="#">1911265</a>	Updated	Programmer	Category B	Complex VPU powerdown might deadlock
<a href="#">1935474</a>	New	Programmer	Category B	Exclusive sequence to 64 bit Peripheral Port does not succeed
<a href="#">1941771</a>	New	Programmer	Category C	DBG_RECOV transition when cluster is OFF might deadlock

**17-Aug-2020: Changes in document version v3.0**

ID	Status	Area	Category	Summary
<a href="#">1905020</a>	New	Programmer	Category B	Interrupt outputs remain asserted after a core is powered off
<a href="#">1911265</a>	New	Programmer	Category B	Complex VPU powerdown might deadlock
<a href="#">1827720</a>	Updated	Programmer	Category B (rare)	PPU can produce spurious interrupt if written to during initialization sequence
<a href="#">1877057</a>	New	Programmer	Category C	Cluster transition to OFF_EMU while core powers on might deadlock
<a href="#">1892065</a>	New	Programmer	Category C	Interconnect DErr on dirty data not reported in RAS registers
<a href="#">1906346</a>	New	Programmer	Category C	PMU events and RAS errors not counted correctly in some 2 node configurations

**12-Jun-2020: Changes in document version v2.0**

ID	Status	Area	Category	Summary
<a href="#">1873142</a>	New	Programmer	Category A	Loss of coherency after exclusive load/store
<a href="#">1814712</a>	New	Programmer	Category B	APB writes can incorrectly update some cluster PMU registers
<a href="#">1869818</a>	New	Programmer	Category B	ATB clock gating might cause deadlock
<a href="#">1827720</a>	New	Programmer	Category B (rare)	PPU can produce spurious interrupt if written to during initialization sequence
<a href="#">1795881</a>	New	Programmer	Category C	Trace flush may not flush all data
<a href="#">1799687</a>	New	Programmer	Category C	Interconnect errors not reported in RAS registers
<a href="#">1836065</a>	New	Programmer	Category C	OFF_EMU to OFF power transitions will deadlock
<a href="#">1852252</a>	New	Programmer	Category C	Incorrect PRIDR0 value
<a href="#">1854639</a>	New	Programmer	Category C	Incorrect RAS CIDR1 value
<a href="#">1861790</a>	New	Programmer	Category C	AXI master PMU events count incorrectly

**31-Mar-2020: Changes in document version v1.0**

ID	Status	Area	Category	Summary
<a href="#">1791722</a>	New	Programmer	Category C	ECC errors in L3 Tag RAMs can cause data corruption and loss of coherency

# Errata summary table

The errata associated with this product affect the product versions described in the following table.

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">2125021</a>	Programmer	Category A	Memory traffic might cause a deadlock	r1p0, r2p0, r2p1	r3p0
<a href="#">2090737</a>	Programmer	Category A	Heavy snoop traffic might cause a deadlock	r1p0, r2p0, r2p1	r3p0
<a href="#">1873142</a>	Programmer	Category A	Loss of coherency after exclusive load/store	r0p0	r1p0
<a href="#">2381090</a>	Programmer	Category B	TLB not invalidated in complex power transitions	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2415004</a>	Programmer	Category B	DSU might lose coherency after data cache clean	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2426404</a>	Programmer	Category B	DVM might violate CHI protocol during cluster powerdown	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2313941</a>	Programmer	Category B	SCLK clock gating might prevent cluster register access	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2309643</a>	Programmer	Category B	CLUSTERL3HIT and CLUSTERL3MISS registers incorrectly decrement	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2304881</a>	Programmer	Category B	Power transition from all slices to one slice might cause deadlock	r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2282666</a>	Programmer	Category B	CLUSTERL3HIT and CLUSTERL3MISS registers count incorrectly	r3p0, r3p1	Open
<a href="#">2091675</a>	Programmer	Category B	Cache debug access while in FUNC_RET might lead to deadlock	r2p0, r2p1	r3p0
<a href="#">2089240</a>	Programmer	Category B	Large number of outstanding ACP read transactions might deadlock	r2p0, r2p1	r3p0
<a href="#">2085140</a>	Programmer	Category B	Debug accesses during core power off might deadlock	r0p0, r1p0, r2p0, r2p1, r3p0	r3p1
<a href="#">2057044</a>	Programmer	Category B	Debug accesses when core powered off might deadlock	r0p0, r1p0, r2p0, r2p1	r3p0
<a href="#">2055944</a>	Programmer	Category B	FUNC_RET while all cores powered off might deadlock	r2p0, r2p1	r3p0
<a href="#">1976458</a>	Programmer	Category B	Cluster PMU or ELA triggers affect core debug	r0p0, r1p0, r2p0	r2p1

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1970782</a>	Programmer	Category B	SCLK gating might cause a deadlock	r2p0	r2p1
<a href="#">1935474</a>	Programmer	Category B	Exclusive sequence to 64 bit Peripheral Port does not succeed	r0p0, r1p0	r2p0
<a href="#">1911265</a>	Programmer	Category B	Complex VPU powerdown might deadlock	r0p0, r1p0	r2p0
<a href="#">1905020</a>	Programmer	Category B	Interrupt outputs remain asserted after a core is powered off	r0p0, r1p0	r2p0
<a href="#">1887098</a>	Programmer	Category B	Isolation incorrectly applied for OFF_EMU power modes	r0p0, r1p0	r2p0
<a href="#">1869818</a>	Programmer	Category B	ATB clock gating might cause deadlock	r0p0	r1p0
<a href="#">1814712</a>	Programmer	Category B	APB writes can incorrectly update some cluster PMU registers	r0p0	r1p0
<a href="#">1827720</a>	Programmer	Category B (rare)	PPU can produce spurious interrupt if written to during initialization sequence	r0p0, r1p0	r2p0
<a href="#">2457882</a>	Programmer	Category C	CLUSTERPMSSRR bits cannot be cleared	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2375351</a>	Programmer	Category C	L3 tag RAM chip enable pins incorrectly asserted in RAM power transitions	r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2330050</a>	Programmer	Category C	SCLK clock gating might prevent cluster register access when PERIPHCLK at a higher frequency	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2349173</a>	Programmer	Category C	MPAMCFG_CPBMS_S_EXCL bit not functioning	r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2301655</a>	Programmer	Category C	Cache debug access might return incorrect data	r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2301453</a>	Programmer	Category C	Incorrect CLUSTERPMU_PMCEID0/IMP_CLUSTERPMCEID0_EL1 value	r0p0, r1p0, r2p0, r2p1, r3p0, r3p1	Open
<a href="#">2261337</a>	Programmer	Category C	PPT_ACCESS access PMU event counts incorrectly for 64-bit AXI Peripheral Port	r0p0, r1p0, r2p0, r2p1, r3p0	r3p1
<a href="#">2165755</a>	Programmer	Category C	Debug APB old CTI address space accessible through new address map	r2p0, r2p1, r3p0	r3p1
<a href="#">2135307</a>	Programmer	Category C	DSU might lose coherency after snoop filter error	r3p0	r3p1
<a href="#">2121479</a>	Programmer	Category C	Debug APB accesses to reserved core address is possible when the core is powered off	r0p0, r1p0, r2p0, r2p1, r3p0	r3p1
<a href="#">2120242</a>	Programmer	Category C	ACP atomic transaction might incorrectly report error	r3p0	r3p1
<a href="#">2049666</a>	Programmer	Category C	L3 cache allocate PMU event counts incorrectly	r0p0, r1p0, r2p0, r2p1	r3p0

ID	Area	Category	Summary	Found in versions	Fixed in version
<a href="#">1985295</a>	Programmer	Category C	Cluster PMU overflow flag might get set when counters are disabled	r0p0, r1p0, r2p0	r2p1
<a href="#">1971432</a>	Programmer	Category C	L3 cache access PMU events count incorrectly	r0p0, r1p0, r2p0	r2p1
<a href="#">1965174</a>	Programmer	Category C	Utility bus accesses to reserved address range might affect other accesses	r0p0, r1p0	r2p0
<a href="#">1961495</a>	Programmer	Category C	Cache debug access while transactions ongoing might lead to deadlock	r2p0	r2p1
<a href="#">1941771</a>	Programmer	Category C	DBG_RECOV transition when cluster is OFF might deadlock	r0p0, r1p0	r2p0
<a href="#">1906346</a>	Programmer	Category C	PMU events and RAS errors not counted correctly in some 2 node configurations	r0p0, r1p0	r2p0
<a href="#">1892065</a>	Programmer	Category C	Interconnect DErr on dirty data not reported in RAS registers	r0p0, r1p0	r2p0
<a href="#">1877057</a>	Programmer	Category C	Cluster transition to OFF_EMU while core powers on might deadlock	r0p0, r1p0	r2p0
<a href="#">1861790</a>	Programmer	Category C	AXI master PMU events count incorrectly	r0p0, r1p0	r2p0
<a href="#">1854639</a>	Programmer	Category C	Incorrect RAS CIDR1 value	r0p0	r1p0
<a href="#">1852252</a>	Programmer	Category C	Incorrect PRIDR0 value	r0p0	r1p0
<a href="#">1836065</a>	Programmer	Category C	OFF_EMU to OFF power transitions will deadlock	r0p0	r1p0
<a href="#">1799687</a>	Programmer	Category C	Interconnect errors not reported in RAS registers	r0p0	r1p0
<a href="#">1795881</a>	Programmer	Category C	Trace flush may not flush all data	r0p0	r1p0
<a href="#">1791722</a>	Programmer	Category C	ECC errors in L3 Tag RAMs can cause data corruption and loss of coherency	r0p0	r1p0

# Errata descriptions

## Category A

2125021

### Memory traffic might cause a deadlock

#### Status

Fault Type: Programmer Category A

Fault Status: Present in r1p0, r2p0, and r2p1. Fixed in r3p0.

#### Description

When there is heavy memory traffic, in some complex cases the DSU might deadlock.

#### Configurations Affected

This erratum affects configurations where NUM\_L3\_SLICES is 4 and NUM\_CORES is greater than 4, or NUM\_L3\_SLICES is 8 for any value of NUM\_CORES. This erratum does not affect Direct Connect configurations.

#### Conditions

The erratum occurs under the following conditions:

1. There is a large number of outstanding transactions between a core and the DSU.
2. Some additional complex micro-architectural timing conditions occur.

#### Implications

If this erratum occurs, then the DSU might deadlock.

#### Workaround

There is no workaround.

## 2090737

### Heavy snoop traffic might cause a deadlock

#### Status

Fault Type: Programmer Category A

Fault Status: Present in r1p0, r2p0, and r2p1. Fixed in r3p0.

#### Description

When there is heavy memory traffic, particularly snoops to multiple cores, in some complex cases the DSU might deadlock.

#### Configurations Affected

This erratum only affects configurations where NUM\_L3\_SLICES is 4 and NUM\_CORES is greater than 4, or NUM\_L3\_SLICES is 8 for any value of NUM\_CORES.

#### Conditions

The erratum occurs under the following conditions:

1. There is a large number of outstanding transactions between the DSU and multiple cores in the cluster.
2. Some additional complex micro-architectural timing conditions occur.

#### Implications

If this erratum occurs, then the DSU might deadlock. This issue has been observed to occur when the slice powerdown feature is in use, as that can generate a high volume of snoop traffic to the cores, however it could also occur in other cases.

#### Workaround

There is no workaround for this erratum.



## 1873142

### Loss of coherency after exclusive load/store

#### Status

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

#### Description

If a core executes a Load-Exclusive or Store-Exclusive instruction, then under certain conditions the DSU might incorrectly transition to a Unique coherency state instead of the correct Shared state. This can lead to a loss of cache coherency with the rest of the system or possibly deadlock.

#### Configurations Affected

This erratum affects configurations of the DSU that meet at least one of the following conditions:

- The DSU has at least one CHI master port and the DSU **BROADCASTOUTER** input pin is HIGH, or
- The DSU has a CHI peripheral port and the DSU **BROADCASTOUTERMP** input pin is HIGH.

In addition, the system must also meet at least one of the following conditions:

- The system interconnect has a snoop filter, or
- There is another fully coherent master connected to the system interconnect.

#### Conditions

1. A Write-Back Cacheable Shareable cache line is cached in the cluster in a Unique state and is cached in at least one core.
2. A core in the cluster executes a Load-Exclusive or Store-Exclusive instruction to the cache line.
3. Some other activity causes a core to evict its copy of the cache line. This causes the DSU to send a ReadPreferUnique(Excl) to the system interconnect to fetch the data for the load or store exclusive.
4. The interconnect returns data in the SC state.

#### Implications

If these conditions are met, along with the listed configurations affected, then the DSU might incorrectly keep the line in a Unique state, while the system interconnect thinks the line is Shared. This can cause loss of coherency for the cache line between the DSU and another master connected to the interconnect, which can lead to data corruption and in some systems potentially deadlock.

#### Workaround

There is no workaround.

## Category A (rare)

There are no errata in this category.

## Category B

2381090

### TLB not invalidated in complex power transitions

#### Status

Fault Type: Programmer Category B.

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

If a core that is part of a complex powers on at the same time that the other core in the same complex is powering off, then the L2 *Translation Lookaside Buffer* (TLB) in the complex might not get invalidated correctly.

#### Configurations Affected

This erratum only affects configurations with two Cortex-A510 cores in a complex.

#### Conditions

1. One core in a complex is in the OFF power mode.
2. The other core in the same complex makes a transition from ON to OFF or OFF\_EMU.
3. At the time the second core has almost completed its power transition, the first core starts a transition from OFF to ON.
4. A third core, outside the complex, executes a TLB invalidate instruction that would invalidate an entry that is currently held in the L2 TLB in the complex.

#### Implications

The L2 TLB in the complex is not invalidated by the power sequences, and so the core retains its previous state. However, there is a brief window in which the complex is disconnected from coherency with the rest of the system and so any TLB invalidate DVMs received during this time will not take effect. This can leave stale entries in the TLB that the cores in the complex might hit when they start executing code again.

#### Workaround

After a core is powered ON, the firmware should execute one of the following sequences before enabling the MMU, depending on whether the system expects to use Secure EL2 or not.

## Without Secure EL2:

```
TLBI ALLE3
Set SCR_EL3.NS=0
ISB
TLBI ALLE1
Set SCR_EL3.NS=1
ISB
TLBI ALLE2
TLBI ALLE1
DSB SY
```

## With Secure EL2:

```
TLBI ALLE3
Set SCR_EL3.NS=0
Set SCR_EL3.EEL2=1
ISB
TLBI ALLE2
TLBI ALLE1
Set SCR_EL3.NS=1
ISB
TLBI ALLE2
TLBI ALLE1
DSB SY
```

## 2415004

### DSU might lose coherency after data cache clean

#### Status

Fault Type: Programmer Category B.

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0 and r3p1. Open

#### Description

If a core executes a DC CVAC, DC CVAP or DC CVADP instruction when the system interconnect is sending a snoop to the cache line and a number of other conditions occur, the DSU might send a clean eviction to the interconnect with stale data.

#### Configurations Affected

This erratum affects configurations that meet all of the following conditions:

- Not a Direct Connect configuration.
- At least one Cortex-A710 or Cortex-X series core is present in the cluster.
- The cluster is connect to a coherent CHI interconnect.
- There is a system cache that is allocated by clean evictions from the cluster.

#### Conditions

This erratum occurs under the following conditions:

1. The DSU register IMP\_CLUSTERECTLR\_EL1 bits [45:44] are non-zero, so the DSU is enabled to send data when evicting clean cache lines. This includes the default value of these bits.
2. A Cortex-A710 or Cortex-X series core executes a DC CVAC, DC CVAP or DC CVADP instruction.
3. The system interconnect sends a snoop to the same cache line of one of the following types. This must occur while condition 2 is in progress. The snoop types that might cause this erratum depend on the value of IMP\_CLUSTERECTLR\_EL1 bits [45:44]:
  - 0x0: The erratum cannot occur.
  - 0x1 or 0x2: SnpCleanShared.
  - 0x3: SnpCleanShared, SnpShared, SnpClean, SnpNotSharedDirty, SnpSharedFwd, SnpCleanFwd, SnpNotSharedDirtyFwd, SnpPreferUnique or SnpPreferUniqueFwd.
4. Additional transactions occur to the same line combined with complex microarchitectural conditions.

#### Implications

If these conditions occur, the DSU might send a WriteEvictOrEvict or WriteEvictFull transaction with stale data. If there is a system cache, it might cache this data. This might cause a loss of coherency.

The data sent with the eviction will always be clean, so it will not cause any problems in a system that does not have a system cache. Also, Arm recommends that systems without a system cache would program IMP\_CLUSTERECTLR\_EL1 bits [45:44] to zero, so such a system would not satisfy condition 1.

## Workaround

If the system will never generate SnpCleanShared to the DSU, firmware should program IMP\_CLUSTERECTLR\_EL1 bits [45:44] to 0x1 or 0x2. This prevents the DSU from sending WriteEvictOrEvict transactions from SC state. In systems without another fully-coherent agent, the SC state will be extremely uncommon, so no performance reduction is expected. In systems with another fully-coherent agent, there might be a reduction in performance of some workloads. Note that a value of 0x0 would also provide a workaround for the erratum, but might cause a larger performance impact.

If the system might generate SnpCleanShared, setting the IMP\_CLUSTERECTLR\_EL1 bits [45:44] to 0x0 would provide a workaround for the issue, but the performance impact is likely to be unacceptable on some workloads. Contact Arm for more details if running on this type of system.

## 2426404

### DVM might violate CHI protocol during cluster powerdown

#### Status

Fault Type: Programmer Category B.

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0 and r3p1. Open

#### Description

If a core executes a *Translation Lookaside Buffer* (TLB) Invalidate or Instruction Cache Invalidate instruction and then the core is powered down without executing a DSB, a subsequent cluster powerdown might violate the CHI protocol.

#### Configurations Affected

This erratum affects all configurations with at least one CHI port, except Direct Connect.

#### Conditions

The erratum occurs under the following conditions:

1. At least one of the following cluster inputs is HIGH:
  - BROADCASTTTLBIINNER
  - BROADCASTTTLBIOUTER
  - BROADCASTICINVAL
2. One core executes a TLB Invalidate or Instruction Cache Invalidate instruction.
3. The cluster does not have any TXREQ credits on the CHI interface to the system interconnect, so it cannot send a DVMOp transaction. This must remain the case during the remaining conditions.
4. That core is powered off without having executed a DSB instruction.
5. The core cache flush does not cause any evictions to be sent to the same CHI interface.
6. The system requests to power off the cluster.
7. The L3 cache flush does not cause any evictions to be sent to the same CHI interface.
8. The DSU requests to disconnect from coherency by driving SYSCOREQ=0 for the CHI ports.

#### Implications

If the erratum occurs, the DSU will send a DVMOp when SYSCOREQ=0, which is a violation of the CHI protocol.

#### Workaround

Firmware should ensure that the powerdown sequence includes a DSB instruction, and does not execute any TLB or Instruction Cache invalidate instruction after that DSB. Arm expects that a typical powerdown sequence would already include a DSB.



## 2313941

### SCLK clock gating might prevent cluster register access

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The SCLK clock domain can be gated when the L3 and the coherency logic are idle. If this gating occurs at the same time that an access to a cluster register is made, then it can be incorrectly gated. This can prevent the access from completing, which might cause a deadlock if there is no further traffic to the SCLK domain from any other core or system component.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The logic in the SCLK domain is idle for at least 256 cycles, and so an internal clock gating sequence starts.
2. An access is made to a cluster register, either as a system register access, a utility bus access, or a debug APB access.
3. The register access happens after the clock gating sequence has started, but before it has completed.
4. No other memory traffic occurs that requires SCLK to be active. This other traffic might be a load or store from a different core that misses in the L1 and L2 caches, an ACP transaction, a snoop transaction from the system interconnect, the **PMUSNAPSHOTREQ** pin is asserted, or there is a cluster power transition.

#### Implications

The clock gating sequence completes, and so SCLK is gated. The **SCLKQACTIVE** signal will be LOW. The register access will not complete until SCLK is ungated again. If no other traffic arrives that causes SCLK to be ungated, then the system might deadlock. In most systems, it is very likely that there will eventually be other traffic that will cause SCLK to be ungated again and therefore avoid the problem. If the FUNC\_RET power mode is enabled then this will significantly reduce the probability of the erratum from occurring because that will cause a power transition after the cluster has been idle for a period of time. However, if the cluster was already in FUNC\_RET when the register access was started then the register access would not bring the cluster out of FUNC\_RET, but it is unlikely that a cluster register access is performed while the cluster is in FUNC\_RET without there being some other related traffic that would cause an exit from FUNC\_RET.

## Workaround

If the system cannot guarantee that there will be other traffic to wake up SCLK, then it should set IMP\_CLUSTERACTLR\_EL1[16:15] bits to 0b11 to disable clock gating of the SCLK domain. This will increase the idle power consumption (typically from less than a few milliwatts to a few tens of milliwatts), and also prevent the external SCLK Q-Channel from allowing SCLK to be gated.

**2309643****CLUSTERL3HIT and CLUSTERL3MISS registers incorrectly decrement****Status**

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

**Description**

The CLUSTERL3HIT and CLUSTERL3MISS registers are typically used to decide when to enter or exit from the L3 RAM powerdown modes. These registers incorrectly decrement the count under certain conditions.

**Configurations Affected**

Configurations with an L3 cache that has more than two slices are affected.

**Conditions**

This erratum occurs under the following conditions:

1. The CLUSTERPWRCTLR.AUTOPTRN field is non-zero, or the CLUSTERL3HIT and CLUSTERL3MISS registers are used by software or firmware.
2. The lower eight bits of the CLUSTERL3HIT or CLUSTERL3MISS register are in the range 0xF8 to 0xFD.
3. Between three and eight cache slices detect an L3 hit or an L3 miss on the same cycle.

**Implications**

When these conditions occur, the CLUSTERL3HIT or CLUSTERL3MISS registers might decrement the value by 256 compared to the true count. This will prevent accurate decisions on when to enter and exit the L3 RAM powerdown modes. The amount of inaccuracy will vary depending on configuration and workloads, but is expected to be less than 20% below the true count, in most cases. This might be acceptable for many systems.

**Workaround**

Some systems might not require a workaround. However, if the level of inaccuracy is too great for the system, then software or firmware should not rely on the values from these registers, and should not set the CLUSTERPWRCTLR.AUTOPTRN field. This might impact the use of the L3 RAM powerdown modes. The cluster PMU can be used to count the L3D\_CACHE and L3D\_CACHE\_REFILL events to generate equivalent information, if the PMU is not in use by other components such as a debugger.

## 2304881

### Power transition from all slices to one slice might cause deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

If the cluster PPU makes an operating mode transition from ALL SLICE to ONE SLICE then, under certain conditions, the cluster might deadlock.

#### Configurations Affected

This erratum affects all configurations with more than one L3 slice configured. It does not affect Direct Connect.

#### Conditions

The erratum occurs under the following conditions:

1. The cluster power operating mode is ALL SLICE.
2. The cluster PPU requests a cluster operating mode transition to ONE SLICE. This will typically be in response to software clearing the IMP\_CLUSTERPWRCTLR\_EL1.SLCRQ bit.
3. During the transition, there is heavy memory traffic from at least one core, from the ACP interface or from the system interconnect snoop requests.

#### Implications

If the above conditions and certain micro-architectural timing conditions occur, a memory transaction from a core or a snoop from the system interconnect might not get a response from the DSU, leading to a deadlock.

#### Workaround

Software should not make use of the ONE SLICE operating mode. It should keep the IMP\_CLUSTERPWRCTLR\_EL1.SLCRQ bit set, which is the default value.

## 2282666

### CLUSTERL3HIT and CLUSTERL3MISS registers count incorrectly

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r3p0 and r3p1. Open.

#### Description

The CLUSTERL3HIT and CLUSTERL3MISS registers are typically used to decide when to enter or exit from the L3 RAM powerdown modes. These registers do not count as many events as they should.

#### Configurations Affected

All configurations with an L3 cache are affected.

#### Conditions

This erratum occurs under the following conditions:

1. The CLUSTERPWRCTLR.AUTOPTRN field is non-zero, or the CLUSTERL3HIT and CLUSTERL3MISS registers are used by software or firmware.

#### Implications

The CLUSTERL3HIT and CLUSTERL3MISS registers will indicate a value significantly below the true count, and so cannot be relied upon. This will prevent accurate decisions on when to enter and exit the L3 RAM powerdown modes.

#### Workaround

Software or firmware should not rely on the values from these registers, and should not set the CLUSTERPWRCTLR.AUTOPTRN field. This might impact the use of the L3 RAM powerdown modes. The cluster PMU can be used to count the L3D\_CACHE and L3D\_CACHE\_REFILL events to generate equivalent information, if the PMU is not in use by other components such as a debugger.

## 2091675

### Cache debug access while in FUNC\_RET might lead to deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r2p0 and r2p1. Fixed in r3p0.

#### Description

The cache debug register provides a method for software at EL3 to directly access the contents of the RAMs inside the DSU for debug purposes. If the cache debug register is accessed while the cluster logic is in the FUNC\_RET power mode, then the DSU might deadlock.

#### Configurations Affected

All configurations, except Direct connect, are affected.

#### Conditions

1. The IMP\_CLUSTERPWRCTLR\_EL1.RETCTL field is set to a non-zero value.
2. The L3 logic is idle for at least the number of architectural timer ticks configured in the IMP\_CLUSTERPWRCTLR\_EL1.RETCTL register, and the cluster is then put into the FUNC\_RET power mode.
3. Software running at EL3 writes and reads from the IMP\_CLUSTERCDBG\_EL3 System register to access one of the DSU RAMs.
4. Additional accesses are made to other cluster registers, or the system requests clock gating on the **SCLK** Q-Channel.

#### Implications

If this erratum occurs, then the cluster might deadlock. The cache debug register is an IMPLEMENTATION DEFINED EL3 register and so only custom EL3 debug software is affected.

#### Workaround

Debug software should:

1. Set the IMP\_CLUSTERPWRCTLR\_EL1.RETCTL field to zero.
2. Execute a DC CIVAC instruction to any address that has a valid page table mapping.
3. Execute a DSB instruction.
4. Perform the required cache debug operations.
5. Restore the previous value of IMP\_CLUSTERPWRCTLR\_EL1.RETCTL once the debug is complete.

## 2089240

### Large number of outstanding ACP read transactions might deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r2p0 and r2p1. Fixed in r3p0.

#### Description

The Accelerator Coherency Port (ACP) interface allows the system to read and write data in the cluster. If a large number of read transactions are outstanding on this interface, then the system can deadlock.

#### Configurations Affected

This erratum only affects configurations with an ACP interface, and either:

- NUM\_L3\_SLICES is 4 and NUM\_LTDBS is 64.
- NUM\_L3\_SLICES is 8 and NUM\_LTDBS is 32 or greater.

#### Conditions

1. The system sends 256 or more read transactions on the ACP interface.
2. The DSU has not yet returned all of the read data for any of the transactions. This implies that the majority of the accesses are missing in the L3 cache.

#### Implications

Data will be returned on the ACP read data channel with the wrong ID value. This could cause data corruption or lead to a system deadlock. The workaround is believed to be acceptable for some classes of usage, however it might not be suitable for all systems.

#### Workaround

If the system can be prevented from sending more than 255 outstanding read transactions, then this erratum will not occur. If the system cannot be limited, then the IMP\_CLUSTERACTLR\_EL1[3:2] should be set to **0b11**. This will limit the number of outstanding reads to 20 which might have a significant impact on the bandwidth available from ACP.

## 2085140

### Debug accesses during core power off might deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r2p0, r2p1 and r3p0. Fixed in r3p1.

#### Description

If an external debugger makes an access on the Debug APB interface to a core that is in the process of powering off, then the access might deadlock. If the **PMUSNAPSHOTREQ** pin is asserted during a similar period, then it might not receive a response.

#### Configurations Affected

All configurations are affected.

#### Conditions

The erratum occurs under the following conditions:

1. A core has started, but not completed, to power off.
2. Either:
  - An access on the Debug APB interface is made to a register in the core.
  - The **PMUSNAPSHOTREQ** pin is asserted.
3. Other traffic is active on the GIC interface, or on the CD APB interface from the cluster to the DebugBlock. If there is back-pressure from the system on these interfaces, then it makes it easier to hit the timing conditions required for this erratum.

Note that a Debug APB access could be an access to any APB register in the core, or to a register in the DebugBlock ROM table or core Cross Trigger Interface (CTI) which are passed through to the core.

#### Implications

If an access is made from the external debugger then it might deadlock, and prevent other cores from powering on. If the **PMUSNAPSHOTREQ** pin is asserted, then it will not complete the handshake but will not impact other transactions.

#### Workaround



If the core PPUs are programmed to set the PPU\_DCDR1.CLKEN\_ISO\_DLY and PPU\_DCDR1.ISO\_RST\_DLY fields to their maximum value, then it will reduce the probability of this erratum occurring. If the system reduces the **PPUCLK** frequency relative to the **PERIPHCLK** frequency, then it would also reduce the probability of the erratum occurring.

## 2057044

### Debug accesses when core powered off might deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r2p0, and r2p1. Fixed in r3p0.

#### Description

If an external debugger makes an access on the Debug APB interface to a core that has recently been powered off, then the access might deadlock.

#### Configurations Affected

This erratum only affects configurations with the configuration parameter SYNC\_LEVELS set to 3.

#### Conditions

The erratum occurs under the following conditions:

1. An access on the Debug APB interface is made while a core is powered on. This could be to a register in any core or the cluster.
2. The core is powered off.
3. A Debug APB access is made to the core.

Note that a Debug APB access could be an access to any APB register in the core, or to a register in the DebugBlock ROM table or core Cross Trigger Interface (CTI) which are passed through to the core.

#### Implications

An access from the external debugger to a core might deadlock if it is the first APB access to the cluster after the core has been powered off.

#### Workaround

The IMP\_CLUSTERACTLR\_EL1[27:26] bits should be set to **2'b11** to disable clock gating of the ATCLK, PCLK, and GICCLK domains. This will increase the idle power consumption in these domains. Note that this register is also accessible as the CLUSTERACTLR register on the utility bus. To avoid increasing power consumption when the debugger is not in use, if the system allows the external debugger to access the utility bus then the external debugger should only set these bits when it is active.

## 2055944

### FUNC\_RET while all cores powered off might deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r2p0 and r2p1. Fixed in r3p0.

#### Description

The DSU can enter the FUNC\_RET low power mode when it has been idle for a certain period of time. If an access arrives while the DSU is in the FUNC\_RET mode, then it will indicate to the PPU that it needs to be moved to the ON power mode to be able to complete the access. If a snoop arrives when all the cores are powered OFF, and ACP is idle, then a snoop will not correctly cause the request to be made for the ON power mode, which might cause a deadlock.

#### Configurations Affected

This erratum affects all configurations except Direct Connect.

#### Conditions

1. The IMP\_CLUSTERPWRCTLR\_EL1.RETCTL field is set to a non-zero value.
2. The IMP\_CLUSTERPWRDN\_EL1.PWRDN field is not set
3. All the cores are powered off.
4. There is no ACP traffic and the **SYSCOREQS** signal is LOW.
5. The L3 logic is idle for at least the number of architectural timer ticks configured in the IMP\_CLUSTERPWRCTLR\_EL1.RETCTL register, and the cluster is then put into the FUNC\_RET power mode.
6. The interconnect sends a snoop to the cluster. This can be any type of snoop, except a DVM message.
7. The cluster PPU has the PPU\_PWPR.PWR\_DYN\_EN bit set to 0, or the PPU\_PWPR.PWR\_POLICY is set to FUNC\_RET.

#### Implications

If the cluster is in FUNC\_RET, the snoop will not complete until the cluster moves to the ON power mode. However, the cluster will incorrectly indicate to the PPU on **DEVPACTIVE** that it requires to be in the OFF mode. If the PPU remains in FUNC\_RET because the requested OFF is below its programmed minimum power mode then the snoop will not complete, which can cause the system to deadlock.

#### Workaround

If the minimum power mode has been set to FUNC\_RET to prevent the cluster from powering off then the IMP\_CLUSTERPWRDN\_EL1.PWRDN bit should be set instead which provides the same functionality. If this bit cannot be used then the FUNC\_RET power mode should be disabled before the last core in the cluster is powered off. This can be done by setting the IMP\_CLUSTERPWRCTLR\_EL1.RETCTL field to 0b000, which is the default value.

## 1976458

### Cluster PMU or ELA triggers affect core debug

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r2p1.

#### Description

The DSU supports a cluster level Performance Monitoring Unit (PMU) as well as an integrated ELA-600 component in the cluster. Both are connected to the Cross Trigger Interface (CTI) logic. If a trigger is generated from either of these components, then it can incorrectly affect the behavior of the core debug logic.

#### Configurations Affected

This erratum only affects configurations with the cluster Embedded Logic Analyzer (ELA) present.

#### Conditions

1. A core either:
  - Leaves reset and starts executing instructions
  - Enters or leaves Debug state
  - Generates a trigger that is sent to the core CTI. The trigger could be from the core debug logic, the Embedded Trace Macrocell (ETM), core PMU, core ELA, or Statistical Profiling Extension (SPE).
2. At the same time, either:
  - The cluster ELA generates a trigger, or
  - The cluster PMU overflows, which generates a trigger.

#### Implications

If the core was leaving reset, then it will deadlock and not start executing any instructions.

If the core was entering or leaving Debug state, then the DBGACK output signal on the DebugBlock might not reflect the correct status of the core. In some cases the core might also deadlock.

If the core was generating a trigger, then this trigger might be dropped before it reaches the CTI, so any cross trigger events will not occur. In some cases the core might also deadlock.

#### Workaround

The cluster ELA should not be programmed to generate any trigger.  
The cluster PMU trigger should not be enabled with the CLUSTERPMINTENSET register, which will also mean that there is no generation of an interrupt on overflow.

## 1970782

### SCLK clock gating might cause a deadlock

#### Status

Fault Type: Programmer Category B  
Fault Status: Present in r2p0. Fixed in r2p1.

#### Description

The SCLK clock domain can be gated when the L3 and coherency logic is idle. If this gating occurs at the same time as a particular sequence of transactions starts, then it can cause a deadlock.

#### Configurations Affected

This erratum affects all configurations except Direct connect.

#### Conditions

1. The L3 logic is idle for at least 256 cycles, and so an internal clock gating sequence starts.
2. A core in either the same cluster or a different cluster executes a DSB instruction. This causes a DVM Sync snoop to be sent to a core in the cluster.
3. Before the core receives the DVM Sync, it starts a transaction that will eventually be sent to L3.
4. Both the transaction from the core and the DVM Sync happen after the clock gating sequence has started, but before it has completed.
5. Additional microarchitectural conditions occur that prevent the core transaction from completing.

#### Implications

The internal clock gating sequence does not complete, which causes a deadlock.

#### Workaround

Set IMP\_CLUSTERACTLR\_EL1[16:15] bits to 0b11 to disable clock gating of the SCLK domain. This will increase the idle power consumption.

**1935474**

## Exclusive sequence to 64 bit Peripheral Port does not succeed

### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

### Description

If a core executes a Load-Exclusive and Store-Exclusive instruction sequence to an address range that matches the 64-bit Peripheral port then it always exclusively fails.

### Configurations Affected

This erratum only affects configurations of the DSU that have an AXI Peripheral Port configured to be 64 bits wide.

### Conditions

1. A Load-Exclusive instruction is executed to an address that maps to the Peripheral Port.
2. A Store-Exclusive instruction is executed to the same address.

### Implications

The AXI ID used for the read transaction sent to the component connected to the Peripheral port is different from the ID used by the write transaction. This can cause the component to treat the accesses as if they were from different sources and therefore always returns an exclusive fail response to the write transaction.

This can cause the software to livelock as the exclusive sequence never successfully completes. Interrupts can still be taken.

### Workaround

If the component connected to the Peripheral port supports atomic transactions, then software can use an atomic instruction instead of an exclusive sequence.



## 1911265

### Complex VPU powerdown might deadlock

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

If a core that is part of a complex has implemented support for powering down the shared VPU logic, then a power transition might deadlock if the other core in the complex makes another power transition at the same time. The VPU isolation might not be correctly applied.

#### Configurations Affected

This erratum only affects configurations with two cores in a complex.

#### Conditions

1. VPU powerdown is enabled in the complex.
2. One core in a complex transitions from ON to FUNC\_RET.
3. During this transition, the other core in the complex also transitions, either from ON to OFF, or from OFF to ON.

#### Implications

The transition of the second core will not complete until the first core makes a new power transition. If the first core does not make a further transition then the system might deadlock.

The **VPUIsolate** output might be incorrectly deasserted in some cases, preventing the VPU power domain from being isolated correctly when powered down.

#### Workaround

FUNC\_RET support should not be enabled on the cores. This can be done by setting IMP\_CPUPWRCTLR\_EL1.VPU\_PWR\_CTRL to 0b000 which is the default value. This will prevent the VPU from entering the FUNC\_RET low power mode.

## 1905020

### Interrupt outputs remain asserted after a core is powered off

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

The interrupt outputs from the core should be deasserted when the core is powered off because the source of the interrupt is no longer active. However, if the source was active and the interrupt was not serviced before the core was powered off, then these interrupt pins remain incorrectly asserted.

#### Configurations Affected

All configurations are affected.

#### Conditions

1. A core is in the ON power mode.
2. The core has at least one interrupt output request pending. This causes one or more of the output pins **nCNTPSIRQ**, **nCNTPSIRQ**, **nCNTVIRQ**, **nCNTHVIRQ**, **nCNTHPIRQ**, **nCNTHSVSIRQ**, **nCNTHPSIRQ**, **nTRBIRQ**, **nPMBIRQ**, **nCOMMIRQ**, **nPMUIRQ**, or **nVCPUMNTIRQ** to be asserted LOW.
3. The core is powered off before the interrupt is serviced.

#### Implications

This might result in the interrupt handler receiving spurious interrupts.

The interrupt outputs will remain asserted when the core is powered off and continue to be asserted when the core is powered on again. They will only be cleared when either the cluster is powered off, or when the same core asserts one of these interrupt sources again.

#### Workaround

When the core is powered on, software at EL3 should program the source of one of these pins to generate an interrupt before it unmask interrupts in the PSTATE register.

## 1887098

### Isolation incorrectly applied for OFF\_EMU power modes

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

If a debugger has requested the use of the Emulated off power modes, then when entering these modes, the isolation cells on the power domain boundaries will be incorrectly enabled. This might cause a deadlock when the debugger tries to access the core.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The debugger programs DBGPRCR.CORENPDRQ to 0b1 in a core to indicate that it does not want power removed from the core.
2. Software running on the core sets the CPUPWRCTLR.CORE\_PWRDN\_EN bit and executes a WFI instruction to start a powerdown sequence. The core will enter the OFF\_EMU power mode.

#### Implications

If the core enters OFF\_EMU, then the **COREISOLATE<sub>n</sub>** or **COMPLEXISOLATE<sub>n</sub>** signals will be incorrectly asserted LOW. If the cluster enters OFF\_EMU then the **CLUSTERISOLATE<sub>n</sub>** signal will be incorrectly asserted LOW. This will prevent any access on the debug APB interface from the debugger to the core or cluster, causing a system deadlock.

#### Workaround

The debugger should not set the DBGPRCR.CORENPDRQ bit and the PPU should not be programmed to enter the OFF\_EMU power mode.

## 1869818

### ATB clock gating might cause deadlock

#### Status

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

#### Description

The ATB trace output from the cluster is in the ATCLK clock domain, which can be clock gated when nothing in the cluster is generating trace data. In some conditions, this clock gating can prevent new trace from a core from being accepted by the cluster, preventing trace output and preventing that core from entering a low power state.

#### Configurations Affected

All configurations are affected.

#### Conditions

1. The ETM in a core, or an ELA in a core or the cluster is generating trace output.
2. The ETM or ELA is disabled, therefore, it stops generating trace data, or the core executes a WFI or WFI instruction or powers off.
3. The ATCLK domain is gated. This can be due to an external request on the ATCLK Q-Channel interface, or due to an internal request based on the clock domain being idle for more than 16 cycles.
4. The same core or a different core starts generating trace data.

#### Implications

The trace data will not be output from the cluster, and the core that generates trace output might stall if it executes a WFI or WFE instruction.

#### Workaround

The IMP\_CLUSTERACTLR\_EL1[27:26] bits should be set to 2'b11 to disable clock gating of the ATCLK, PCLK, and GICCLK domains. This will increase the idle power consumption in these domains.

## 1814712

### APB writes can incorrectly update some cluster PMU registers

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

Writes to CLUSTERPMU\_PMCCNTR and CLUSTERPMU\_PMEVCNTR<x> can corrupt the value in the register when accessed over the debug APB interface.

#### Configurations Affected

This erratum affects all configurations of the DSU.

#### Conditions

A write is made on the debug APB bus to either half of the CLUSTERPMU\_PMCCNTR register, or to the upper half of any of the CLUSTERPMU\_PMEVCNTR<x> registers.

#### Implications

After the write has completed, parts of the register contain an incorrect value. This prevents using write accesses to these registers from the APB interface. System register access to them is unaffected.

#### Workaround

If write access is required to these registers, then system register access should be used instead of memory-mapped access.

## Category B (rare)

1827720

### PPU can produce spurious interrupt if written to during initialization sequence

#### Status

Fault Type: Programmer Category B Rare

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

When the cluster **nRESET** pin is deasserted, the cluster and core PPUs perform an initialization transition to OFF, including a PCSM handshake. If a write to the PPU\_PWPR register occurs while the initialization transition is ongoing, then a spurious static full, power, or operating policy transition completion interrupt can be asserted under certain conditions, or a static operating policy transition completion interrupt may not be asserted when it should.

#### Configurations affected

All configurations are affected.

#### Conditions

##### Static full policy transition completion interrupt

If a write to the PPU\_PWPR register occurs with the following conditions, then a spurious static full policy transition completion interrupt can be asserted.

- The write occurs to the PPU\_PWPR register before the initialization transition is complete with all the following conditions:
  - Any dynamic mode transition enable is set to 1 (PWR\_DYN\_EN = 1 or OP\_DYN\_EN = 1).
  - The power policy (PWR\_POLICY) is set to OFF.
  - If the PPU written is the cluster PPU, then the operating mode policy (OP\_POLICY) is set to ONE SLICE SFONLY.

##### Static power policy transition completion interrupt

If a write to the PPU\_PWPR register occurs with the following conditions, then a spurious static power policy transition completion interrupt can be asserted.

- The static power policy transition completion interrupt is unmasked:
  - PPU\_AIMR.STA\_POLICY\_PWR\_IRQ\_MASK = 0b0
- A write occurs to the PPU\_PWPR register before the initialization transition is complete with all the

following conditions:

- The power mode dynamic transition enable is set to 1 (PWR\_DYN\_EN = 0b1).
- If the PPU written is the cluster PPU then the operating mode policy (OP\_POLICY) is set to ONE SLICE SFONLY.

### Static operating policy transition completion interrupt

If a write to the PPU\_PWPR register occurs with the following conditions, then the static operating mode policy transition completion interrupt is not correctly asserted.

- The static power policy transition completion interrupt is unmasked:
  - PPU\_AIMR.STA\_POLICY\_OP\_IRQ\_MASK = 0b0
- A write occurs to the PPU\_PWPR register before the initialization transition is complete with all the following conditions:
  - The operating mode dynamic transition enable is set to 1 (OP\_DYN\_EN = 0b1).
  - If the PPU written is the cluster PPU then the operating mode policy (OP\_POLICY) is set to ONE SLICE SFONLY.

If the operating mode dynamic transition enable is set to 1 (OP\_DYN\_EN = 0b1) then the interrupt is asserted incorrectly, if it is set to 0 then the interrupt should be generated but is not asserted.

## Implications

A spurious interrupt is generated on the **CLUSTERPPUIRQ** or **COREPPUIRQ** pins, or an intended interrupt is not generated.

The time in which the interrupt output is incorrect is limited to a period following a reset deassertion. This period can be very short. Unless an access is made to the PPU immediately following the reset, it is unlikely to traverse any interconnect and reach the PPU before this period is expired, especially considering it would typically have to traverse at least one asynchronous boundary.

## Workaround

This workaround involves getting the PPU to produce a known interrupt, which indicates that past that point, no further spurious interrupts can be sent.

- Leave the power policy and operating policy transition completion interrupts masked.
  - PPU\_AIMR.STA\_POLICY\_PWR\_IRQ\_MASK = 0b1 (Default)
  - PPU\_AIMR.STA\_POLICY\_OP\_IRQ\_MASK = 0b1 (Default)
- Read the PPU\_PWPR register.
- AND the read value of the PPU\_PWPR register with 0xFEFFFFFF.
- Write this value back to the PPU\_PWPR register.
- Wait for the *static full policy transition completion interrupt*.
- Clear the IRQ status for this interrupt by writing PPU\_ISR with value 0x0000\_0001.
- Check for any other valid interrupt statuses and handle as appropriate.
- Write the PPU\_PWPR register back to the initial read value.
- Then, continue normal operation. At this point the power policy and operating policy transition

completion interrupts can be unmasked, if the interrupts are required.



## Category C

2457882

### CLUSTERPMSSRR bits cannot be cleared

#### Status

Fault Type: Programmer Category C.

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The CLUSTERPMSSRR register controls whether the PMU counters are reset after a snapshot is taken. If this register is written to set any of the bits, then those bits will be set, but further writes to the register will be unable to clear them.

#### Configurations Affected

This erratum affects all configurations except Direct connect.

#### Conditions

The erratum occurs under the following conditions:

1. The CLUSTERPMSSRR register is written with at least one of the RP bits set.
2. The CLUSTERPMSSRR register is written to clear one of the RP bits that is already set.

#### Implications

The RP bits that are set will not be cleared. The snapshot feature will work when the reset of the counters enabled is or disabled, however the software will not be able to switch between the two modes more than once, until the cluster is reset.

#### Workaround

The software should chose which mode it requires and then avoid subsequent switches to the other mode.

## 2375351

### L3 tag RAM chip enable pins incorrectly asserted in RAM power transitions

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The DSU supports two low-power operating modes which can be used independently or together at the same time. If they are used together and there is a transition from ONE SLICE mode to ALL slice mode while the RAMs are in the 1/2 RAM or SFONLY modes, then the L3 tag RAM invalidation performed as part of the ONE SLICE to ALL SLICE transition gets incorrectly applied to the RAM ways that are powered down as well as the intended RAMs.

This might violate the timing or functional requirements of some RAMs.

#### Configurations Affected

This erratum affects configurations of the DSU with an L3 cache, and with the L3 cache RAM Power down mode implemented.

#### Conditions

1. The cluster power operating mode is ONE SLICE 1/2 RAM or ONE SLICE SFONLY.
2. The operating mode is changed to ALL SLICE 1/2 RAM or ALL SLICE SFONLY.
3. The RAMs used by the implementation require the clock to be gated and the chip enable pins to be deasserted when they are powered off.

#### Implications

The physical requirements of the L3 tag RAMs might be violated. The effects will depend on the specific RAM implementation.

When the RAMs are later powered on, they will be written to invalidate them before use, so there is no logical problem unless the physical effects persist after the power is restored.

#### Workaround

Arm expects that the majority of RAM implementations do not have any persistent effect from this situation, therefore no workaround is required.

If a workaround is required, then the software should ensure that either the RAM Power down modes or the slice Power down modes are used, but not both at the same time.

## 2330050

### SCLK clock gating might prevent cluster register access when PERIPHCLK at a higher frequency

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The SCLK clock domain can be gated when the L3 and the coherency logic are idle. If this gating occurs at the same time that an access to a cluster register is made, then it can be incorrectly gated. This can prevent the access from completing, which might cause a deadlock if there is no further traffic to the SCLK domain from any other core or system component.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The logic in the SCLK domain is idle for at least 256 cycles, and so an internal clock gating sequence starts.
2. An access is made to a cluster register, either as a system register access, a utility bus access, or a debug APB access.
3. The register access happens after the clock gating sequence has started, but before it has completed.
4. No other memory traffic occurs that requires SCLK to be active. This other traffic might be a load or store from a different core that misses in the L1 and L2 caches, an ACP transaction, a snoop transaction from the system interconnect, the **PMUSNAPSHOTREQ** pin is asserted, or there is a cluster power transition.
5. The **PERIPHCLK** domain is running at a higher clock frequency than the **SCLK** domain.

#### Implications

The clock gating sequence completes, and so SCLK is gated. The **SCLKQACTIVE** signal will be LOW. The register access will not complete until SCLK is ungated again. If no other traffic arrives that causes SCLK to be ungated, then the system might deadlock. In most systems, it is very likely that there will eventually be other traffic that will cause SCLK to be ungated again and therefore avoid the problem. If the FUNC\_RET power mode is enabled then this will significantly reduce the probability of the erratum from occurring because that will cause a power transition after the cluster has been idle for a period of time. However, if the cluster was already in FUNC\_RET when the register access was started then the register access would not bring the cluster out of FUNC\_RET, but it is unlikely that a cluster register access is performed while the cluster is in FUNC\_RET without there being some other related traffic that would cause an exit from FUNC\_RET.

## Workaround

Arm expects that in typical operation PERIPHCLK will be running slower than SCLK. For such systems, no workaround is needed.

If the system can run SCLK slower than PERIPHCLK, and cannot guarantee that there will be other traffic to wake up SCLK, then it should set IMP\_CLUSTERACTLR\_EL1[16:15] bits to 0b11 to disable clock gating of the SCLK domain. This will increase the idle power consumption (typically from less than a few milliwatts to a few tens of milliwatts), and also prevent the external SCLK Q-Channel from allowing SCLK to be gated.

## 2349173

### MPAMCFG\_CPBM\_s.S\_EXCL bit not functioning

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The DSU has an IMPLEMENTATION DEFINED field, MPAMCFG\_CPBM\_s.S\_EXCL for the cache partitioning that allows Secure software to control which portions of the cache are unavailable for Non-secure software. This bit does not get transferred correctly within the DSU and therefore does not take effect.

#### Configurations Affected

This erratum affects all configurations, except Direct connect.

#### Conditions

The erratum occurs if all the following conditions apply:

1. The Secure software uses the cache partitioning, and uses it to reserve a part of the cache for the exclusive use of Secure software.
2. The MPAMCFG\_CPBM\_s.S\_EXCL bit is set for one or more of the partitions.

#### Implications

Non-secure software can allocate cache lines in the ways that Secure software had reserved for itself. This does not impact the security of those lines, but it means that the performance of the Secure software could be lower than expected.

#### Workaround

No workaround is required for this erratum.

## 2301655

### Cache debug access might return incorrect data

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The cache Debug register provides a method for software at EL3 to directly access the contents of the RAMs inside the DSU for debug purposes. If the cache Debug register is accessed while there are other memory transactions accessing the DSU, then in rare cases, incorrect data might get returned by the cache Debug register.

#### Configurations Affected

This erratum affects all configurations, except Direct connect.

#### Conditions

The erratum occurs if all the following conditions apply:

1. Software running at EL3 writes and reads from the IMP\_CLUSTERCDBG\_EL3 System register to access one of the DSU RAMs.
2. A large number of memory transactions are made to the DSU from other cores, from the system interconnect, or from a DSU power transition.

#### Implications

If this erratum occurs, then incorrect data will be read from the cache Debug register. The cache Debug register is an IMPLEMENTATION DEFINED EL3 register, and therefore only custom EL3 debug software is affected.

Typical uses of the cache Debug register would already have other cores idle while reading the cache contents; to avoid disturbing the cache contents. Therefore, other memory transactions are unlikely to occur in such uses.

#### Workaround

Before executing cache debug instructions, debug software should ensure that other cores and system components that might access the DSU are idle.

## 2301453

### Incorrect CLUSTERPMU\_PMCEID0/IMP\_CLUSTERPMCEID0\_EL1 value

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, r3p0, and r3p1. Open.

#### Description

The CLUSTERPMU\_PMCEID0/IMP\_CLUSTERPMCEID0\_EL1 register indicates which PMU events are implemented. It incorrectly indicates that the CHAIN is implemented.

#### Configurations Affected

This erratum affects all configurations, except Direct connect.

#### Conditions

The CLUSTERPMU\_PMCEID0/IMP\_CLUSTERPMCEID0\_EL1 register is read and the contents of bit[30] is used.

#### Implications

The software trying to discover the available PMU events might use the CHAIN event which is not implemented.

#### Workaround

The software should ignore the value of the incorrect bit.

## 2261337

### PPT\_ACCESS access PMU event counts incorrectly for 64-bit AXI Peripheral Port

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r2p0, r2p1 and r3p0. Fixed in r3p1.

#### Description

The cluster PMU implements various performance monitor events. If the PMU is programmed to count the implementation-defined PPT\_ACCESS event, 0x0219, then in certain cases it might fail to count an event.

#### Configurations Affected

All configurations with a 64-bit AXI Peripheral Port are affected.

#### Conditions

This erratum occurs under the following conditions:

1. One of the PMU event counters is configured to count the PPT\_ACCESS event, 0x0219.
2. The AXI peripheral port sends a 64-byte read transaction to the system.
3. On the same cycle, the AXI peripheral port sends a 64-byte write transaction to the system.

#### Implications

If this erratum occurs, the counter should increment by 16, but due to this erratum, it does not change value. The PMU counter will be lower than the correct value.

#### Workaround

This erratum has no workaround.



## 2165755

### Debug APB old CTI address space accessible through new address map

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r2p0, r2p1 and r3p0. Fixed in r3p1.

#### Description

The DSU can be configured with either the old or new debug APB address map, using the **DEBUG\_NEW\_ADDR\_MAP** configuration parameter. If the new debug APB address map is configured, and a read or write transaction is sent on the Debug APB interface to an address that would be in the Cross Trigger Interface (CTI) address space in the old debug APB address map, it should be treated as RAZ/WI. Due to this erratum, the transaction does access the CTI space as if the old debug APB address map was configured.

#### Configurations Affected

This erratum affects all configurations of the DSU where **DEBUG\_NEW\_ADDR\_MAP** is **TRUE**.

#### Conditions

The erratum occurs under the following conditions:

1. A read or write transaction is sent on the Debug APB interface.
2. The transaction is to an address in the range 0x10000 - 0xBFFFF.

#### Implications

The CTI address space is mapped twice, once at the old address range and again at the new address range. Accesses to either the old or new address space to the CTI registers will function correctly. Software that is written correctly should not generate accesses to these reserved regions.

#### Workaround

No workaround is necessary.

## 2135307

### DSU might lose coherency after snoop filter error

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r3p0. Fixed in r3p1.

#### Description

The DSU might not detect, correct, or report any snoop filter RAM ECC errors that cause the corrupted entry to look similar to another entry in certain situations. When this happens, a snoop filter entry might be lost, causing both a loss of coherency and data corruption.

#### Configurations Affected

This erratum affects all configurations except Direct Connect.

#### Conditions

This erratum occurs under the following conditions:

1. The DSU reads the snoop filter RAM due to a new CPU, ACP, or snoop transaction.
2. There is at least one snoop filter ECC error, which can be a single-bit error or a double-bit error. This causes two snoop filter entries to appear to contain the same physical address as the transaction.
3. One cycle later, a transaction causes the DSU to read the same snoop filter RAM entries again.
4. All the snoop filter ways for that index are already valid.

#### Implications

If these conditions are met and certain micro-architectural conditions occur, then the DSU might overwrite the contents of a snoop filter entry, losing what was previously there. This could result in both a loss of coherency and data corruption.

The DSU might not report any ECC error, or it might report the snoop filter error in the error record status register, `ERR0STATUS`, as Correctable, or as Uncontainable and Critical.

There is still substantial benefit to be gained from the ECC logic. This erratum might cause a negligible increase in overall system failure rate.

#### Workaround

No workaround is required.

## 2121479

### Debug APB accesses to reserved core address is possible when the core is powered off

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r2p0, r2p1, and r3p0. Fixed in r3p1.

#### Description

If a read or write transaction is made on the Debug APB interface to a reserved address in the core address space, then it should be treated as RAZ/WI while the core is powered on. If certain reserved addresses are accessed, then they will be treated as RAZ/WI even when the core is powered off, instead of giving an error response.

#### Configurations Affected

This erratum affects all configurations of the DSU.

#### Conditions

The erratum occurs under the following conditions:

1. A read or write access is made on the Debug APB interface.
2. The access is to an address:
  - 0x01nm\_0D90 when the **DEBUG\_NEW\_ADDR\_MAP** configuration parameter is FALSE. n is a value from 0 to the number of cores configured minus one. m is any value.
  - 0x00nm\_0D90 when the **DEBUG\_NEW\_ADDR\_MAP** configuration parameter is TRUE. n is a value from 1 to the number of cores configured. m is any value except 0xE.

#### Implications

The software is not expected to access these reserved registers, therefore there are no implications.

#### Workaround

No workaround is necessary.

## 2120242

### ACP atomic transaction might incorrectly report error

#### Status

Fault Type: Programmer Category C  
Fault Status: Present in r3p0. Fixed in r3p1.

#### Description

AtomicCompare transactions on the Accelerator Coherency Port (ACP) might incorrectly report an SLVERR response if the system interconnect returns poisoned data on part of the transaction response.

#### Configurations Affected

This erratum only affects configurations with an ACP interface and a CHI interface to the system interconnect. It also requires a system interconnect that supports poisoning data on atomic transactions.

#### Conditions

The erratum occurs under the following conditions:

1. The **BROADCASTATOMIC** pin or **BROADCASTATOMICMP** pin is HIGH indicating that the interconnect supports atomic transactions on the main or peripheral CHI ports.
2. An AtomicCompare transaction of 16 bytes is sent on the ACP interface.
3. Any of the following conditions are met:
  - a. The AtomicCompare is to device, Non-cacheable, or Write-Through memory
  - b. The AtomicCompare is to Write-Back Non-shareable memory
  - c. The AtomicCompare is to Write-Back Inner Shareable or Outer Shareable memory, and the **BROADCASTOUTER** pin or **BROADCASTOUTERMP** pin is HIGH indicating that the interconnect is coherent on the main or peripheral CHI ports, and the CLUSTERECTLR register bit[7] is zero.
4. The AtomicCompare is passed through the DSU and sent as a CHI AtomicCompare transaction to the system interconnect.
5. The system interconnect returns the data response, and there is a poison bit set in the response but only on bytes of the data that were not requested.

#### Implications

For systems that can generate poison on the unused part of the transaction, this might result in the ACP transaction incorrectly returning an SLVERR response.

In typical systems, the data and poison bits in the unused part of the response would contain data either from an earlier transaction or to an address adjacent to the requested data. In these cases, the poison bits would only be set if there had been a fault detected on a different address. Therefore, there might be a small increase in overall system failure rate because of this erratum.

## Workaround

There is no workaround.

## 2049666

### L3 cache allocate PMU event counts incorrectly

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r2p0, and r2p1. Fixed in r3p0.

#### Description

The cluster PMU implements various performance monitor events. As explained in this erratum, some L3 cache access events will not count correctly under certain conditions.

#### Configurations Affected

All configurations with an L3 cache configured are affected.

#### Conditions

1. One of the PMU event counters is configured to count event 0x0029, L3D\_CACHE\_ALLOCATE.
2. Either:
  - The core is in write streaming mode for a large write stream that is not allocating to L3, but the line being written is already present in the L3 or another core in the cluster.
  - An eviction of clean data from L2, when the line is also present in L3 or another core in the cluster.

#### Implications

The counter value for this event will not increment for the write streaming data that allocates into L3, and will increment for the clean evictions that do not allocate to L3. This will result in minor errors in the counted values.

#### Workaround

No workaround is necessary.

## 1985295

### Cluster PMU overflow flag might get set when counters are disabled

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r2p1.

#### Description

The cluster Performance Monitoring Unit (PMU) implements various performance monitor events, including a cycle counter. If these counters are disabled, then they will not increment, however the overflow flag may still get set.

#### Configurations Affected

All configurations are affected except Direct connect.

#### Conditions

1. An event counter or the cycle counter contains their maximum value of 0xFFFFFFFFFFFFFFFF.
2. The event or cycle counter is disabled.
3. The selected event occurs, or for the cycle counter the next clock cycle occurs.

#### Implications

The event counter or the cycle counter will not be incremented because they are not enabled. However, the bits in the IMP\_CLUSTERPMOVSET\_EL1 and IMP\_CLUSTERPMOVCLR\_EL1 registers will incorrectly get set, indicating an overflow has occurred.

#### Workaround

No workaround is necessary.

## 1971432

### L3 cache access PMU events count incorrectly

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, and r2p0. Fixed in r2p1.

#### Description

The cluster PMU implements various performance monitor events. As explained in this erratum, some L3 cache access events will not count correctly under certain conditions.

#### Configurations Affected

All configurations with an L3 cache configured are affected.

#### Conditions

1. One of the PMU event counters is configured to count any of the following events when the **BROADCASTOUTER** configuration pin is LOW or the main master ports are configured as AXI:
  - 0x002A, L3D\_CACHE\_REFILL
  - 0x00A2, L3D\_CACHE\_REFILL\_RD
  - 0x00A3, L3D\_CACHE\_REFILL\_WR
2. One of the PMU event counters is configured to count any of the following events, and an atomic instruction is executed:
  - 0x002B, L3D\_CACHE
  - 0x00A0, L3D\_CACHE\_RD
  - 0x00A1, L3D\_CACHE\_WR
3. One of the PMU event counters is configured to count any of the following events, and a write to device, non-cacheable, or non-shareable memory is executed:
  - 0x00A1, L3D\_CACHE\_WR
  - 0x0029, L3D\_CACHE\_ALLOCATE

#### Implications

Counter values:

- Set of conditions 1 and 2: The counter values for these events will not increment.
- Set of condition 3: The counter values will count in the conditions listed when they should not.

Consequence:

- Set of condition 1: It might limit the use of the counters in some systems, however these systems are



- not believed to be widely deployed.
- Set of conditions 2 and 3: They will result in minor errors in the counted values.

## Workaround

There is no workaround.

## 1965174

### Utility bus accesses to reserved address range might affect other accesses

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

The Utility bus provides memory-mapped access to various registers in the cluster. If a read and a write occur at the same time, and one of the accesses is to a reserved address region, then it can incorrectly affect the behavior of the other access.

#### Configurations Affected

This erratum affects all configurations of the DSU.

#### Conditions

The erratum occurs under the following conditions:

1. A read access is made on the Utility bus.
2. A write access is made on the Utility bus, at the same time as the read is ongoing.
3. One of the two accesses is to the reserved address region between 0x040000 and 0x07FFFF.

#### Implications

The access to the reserved address region will be treated correctly as RAZ/WI. The other access to a different address will also be treated incorrectly as RAZ/WI.

Software that is written correctly should not generate accesses to this reserved region. If the system allows unprivileged accesses to be routed to the Utility bus, then an unprivileged process could cause incorrect behavior of a privileged process.

This is not expected to be an issue for sample silicon.

#### Workaround

No workaround is necessary.

## 1961495

### Cache debug access while transactions are ongoing might lead to deadlock

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r2p0. Fixed in r2p1.

#### Description

The cache debug register provides a method for software at EL3 to directly access the contents of the RAMs inside the DSU for debug purposes. If the cache debug register is accessed while other transactions are ongoing, then the DSU might deadlock.

#### Configurations Affected

All configurations except Direct connect are affected.

#### Conditions

1. Software running at EL3 writes to the IMP\_CLUSTERCDBG\_EL3 System register to access the L3 Data RAM. The write value must have bits [1:0]=0b11.
2. During this time, one of the following conditions occurs:
  - Software running on a core in the cluster executes an explicit load or store instruction to Device non-Reorderable memory, or
  - A core in the cluster is performing a Secure memory system transaction to the DSU with a physical address that falls within a specific 64-byte range. Bits [17:6] of the physical address must match bits [17:6] of the value written to IMP\_CLUSTERCDBG\_EL3. Bits [18] and above of the physical address must be 0. This could include speculative accesses, instruction fetches, or pagewalks.

#### Implications

If the erratum occurs, then the cluster might deadlock. The cache debug register is an IMPLEMENTATION DEFINED EL3 register and so only custom EL3 debug software is affected. Cache debug operations would normally only be used when the caches are disabled, to ensure that a consistent state is observed. This reduces the probability of other transactions occurring at the same time.

#### Workaround

No workaround is necessary.

## 1941771

### DBG\_RECOV transition when cluster is OFF might deadlock

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

The DBG\_RECOV power mode is provided to allow reset of the cluster while preserving the L3 cache RAM contents for debugging. If the DBG\_RECOV mode is requested as the first transition after the whole cluster is reset with the nRESET pin, then it might deadlock.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The nRESET pin is asserted and then deasserted to perform a cold reset of the entire cluster including the PPUs.
2. The PPU\_PTCR.DBG\_RECOV\_PORST\_EN bit is not set.
3. The DBG\_RECOV power mode is requested in the cluster PPU.

#### Implications

The system will deadlock, and the cache contents cannot be recovered for debug purposes.

DBG\_RECOV can still be used from other power modes such as ON.

Note that the deadlock could occur in any case where DBG\_RECOV is requested from the OFF state.

However, if the cluster is actually powered off, then the RAM contents would already be lost and therefore it would not be useful to request DBG\_RECOV mode.

#### Workaround

After the nRESET pin has been deasserted, the following sequence should be used:

1. The PPU\_PTCR.DBG\_RECOV\_PORST\_EN bit should be set in the cluster PPU. This will apply a cold reset during DBG\_RECOV, however that will not cause any additional loss of state because the logic will already have been cold reset by the assertion of the nRESET pin.
2. Program the cluster PPU to go to DBG\_RECOV, and wait for the transition to complete.
3. Program the cluster PPU to go to ON, and wait for the transition to complete.
4. Clear the PPU\_PTCR.DBG\_RECOV\_PORST\_EN bit in the cluster PPU.

5. Program the cluster PPU to go to DBG\_RECOV, and wait for the transition to complete.
6. Program the core PPUs to go to DBG\_RECOV, and wait for the transition to complete.
7. Program the cluster PPU to go to ON.
8. Program the core PPUs to go to ON.

## 1906346

# PMU events and RAS errors not counted correctly in some 2 node configurations

## Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

## Description

The L3 cache is implemented as several cache slices. Each cache slice can generate some PMU events and detect RAS errors, both of which are collated in a central location. In some configurations, these events are not correctly sent to the central location from some of the cache slices.

## Configurations Affected

This erratum only affects configurations with `TRANSPORT_HORIZ_REG_SLICE` set to 1 or 2, and either of the following conditions, which give a 2 node transport configuration:

- `NUM_L3_SLICES` is 4 and up to 2 cores (or 4 cores if they are combined in 2 complexes).
- `NUM_L3_SLICES` is 2 and between 3 and 4 cores (or 6 to 8 cores if they are combined into 3 or 4 complexes).

## Conditions

1. A PMU event or ECC error occurs in L3 cache slice 1 (in a 2 slice configuration) or in L3 cache slices 2 or 3 (in a 4 slice configuration).

## Implications

PMU events from these L3 cache slices will not be counted. ECC errors detected in these L3 cache slices will not be reported in the RAS registers. The error will still be corrected or cause data poisoning as normal.

## Workaround

There is no workaround.

## 1892065

### Interconnect DErr on dirty data not reported in RAS registers

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

Some errors reported by the interconnect on linefills that allocate to L3 will not be reported in the DSU RAS Error Record Registers.

#### Configurations Affected

This erratum affects configurations of the DSU with at least one CHI master port or CHI peripheral port configured. It does not affect Direct connect configurations.

It also requires the interconnect to use DErr responses rather than the poison support on CHI.

#### Conditions

1. The DSU requests a linefill that allocates to L3. This could be generated from a PRFM instruction targeting L3 or, on some cores, the hardware prefetcher can generate these requests.
2. The interconnect returns dirty data with a DErr response indicating that there is an error in the data.

#### Implications

The DSU will not allocate the line to L3 because of the error, and so the dirty data will be discarded. The DSU RAS Error Record Registers are not correctly updated to indicate that the dirty data was lost. The original cause of the error happened outside of the DSU and should have been logged by the component that detected the error, although, this might have been recorded as a deferred error. For systems that use DErr responses, there might be a negligible increase in overall system failure rate because of this erratum. However, any system where RAS is particularly important would be expected to use the poison field rather than signal a DErr, since the poison allows the line to be allocated into L3 and the error can remain deferred.

#### Workaround

No workaround is necessary.

## 1877057

### Cluster transition to OFF\_EMU while core powers on might deadlock

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

If a debugger has requested the use of the emulated off power modes, then this can cause a deadlock if the cores power off and on while the cluster is also in the middle of a power transition.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The debugger programs DBGPRCR.CORENPDRQ to 0b1 in at least one core to indicate that it does not want power removed from the core.
2. Software running on the core sets the CPUPWRCTLR.CORE\_PWRDN\_EN bit and executes a WFI instruction to start a powerdown sequence. The core will enter the OFF\_EMU power mode.
3. All other cores in the cluster are also in the OFF\_EMU mode or OFF. This causes the cluster to start to enter the OFF\_EMU power mode.
4. One of the cores is powered on. This could be from OFF to ON or OFF\_EMU to ON.
5. The core power on starts in a short period while the cluster transition is still in progress.

#### Implications

If the debugger is active when software powers the cores off and on in close succession, then the core might power ON while the cluster remains in OFF\_EMU. This can cause a deadlock.

#### Workaround

If a workaround is required, then the debugger should set the IMP\_CLUSTERPWRDN\_EL1.PWRDN bit before it sets the DBGPRCR.CORENPDRQ bit. This will keep the cluster in the ON mode rather than going to OFF\_EMU.



## 1861790

### AXI master PMU events count incorrectly

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0 and r1p0. Fixed in r2p0.

#### Description

The cluster PMU implements various performance monitor events. Because of this erratum, the AXI master port and peripheral port events will not count correctly.

#### Configurations Affected

Only configurations with an AXI master port or AXI peripheral port present are affected.

#### Conditions

One of the PMU event counters is configured to count any of the following events when the main master ports are configured as AXI:

- 0x0019, BUS\_ACCESS
- 0x0060, BUS\_ACCESS\_RD
- 0x0061, BUS\_ACCESS\_WR
- 0x0062, BUS\_ACCESS\_SHARED
- 0x0063, BUS\_ACCESS\_NOT\_SHARED
- 0x0064, BUS\_ACCESS\_NORMAL
- 0x0065, BUS\_ACCESS\_PERIPH

One of the PMU event counters is configured to count any of the following events when the peripheral port is configured as AXI:

- 0x0219, PP\_ACCESS
- 0x0260, PP\_ACCESS\_RD
- 0x0261, PP\_ACCESS\_WR

#### Implications

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

#### Workaround

There is no workaround.

## 1854639

### Incorrect RAS CIDR1 value

#### Status

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

#### Description

The CIDR1 register in the memory mapped RAS registers contains an incorrect value in the CLASS field.

#### Configurations affected

All configurations are affected.

#### Conditions

Software reads the memory mapped CIDR1 register at offset 0x20FF8 on the utility bus.

#### Implications

The CIDR1.CLASS field incorrectly has the value 0x9 indicating a CoreSight Component, instead of the expected value of 0xF defined by the RAS architecture. Any software that uses this field to identify the component might not correctly identify it as RAS component.

#### Workaround

Software should ignore the value of this field.

## 1852252

### Incorrect PRIDR0 value

#### Status

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

#### Description

The Power Request Identification Register 0 (PRIDR0) register in the cluster ROM table indicates which power request functionality is present. This register will incorrectly have the SYSRR and DBGRR bits set.

#### Configurations affected

All configurations are affected.

#### Conditions

The external debugger reads the PRIDR0 register and uses the contents of the SYSRR or DBGRR fields (bits [5:4]).

#### Implications

The PRIDR0 contents incorrectly indicate that the reset functionality is implemented. Any writes to the DBGRSTRR, DBGRSTAR, SYSRSTRR, and SYSRSTAR registers will have no effect.

#### Workaround

The debugger should not attempt to make use of these features.

## 1836065

### OFF\_EMU to OFF power transitions will deadlock

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

A cluster power transition from OFF\_EMU to OFF will deadlock.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. The debugger programs DBGPRCR.CORENPDRQ to 0b1 in at least one core to indicate it does not want power removed from the core.
2. Software running on the core sets the CPUPWRCTLR.CORE\_PWRDN\_EN bit and executes a WFI instruction to start a powerdown sequence. The core will enter the OFF\_EMU power mode.
3. All other cores in the cluster are also in the OFF\_EMU mode or OFF. This causes the cluster to enter the OFF\_EMU power mode.
4. The debugger clears the DBGPRCR.CORENPDRQ bit and all cores enter the OFF power mode. This will cause the cluster to attempt to transition from OFF\_EMU to OFF.

#### Implications

If the debugger is active when software powers down the cores, and then the debugger finishes debugging before the cores power up again, the cluster will not power down and will cause a system deadlock.

#### Workaround

If a workaround is required, then once the debugger has set the DBGPRCR.CORENPDRQ bit it should not clear it. This will keep the cluster in the OFF\_EMU mode rather than entering OFF.

## 1799687

### Interconnect errors not reported in RAS registers

#### Status

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

#### Description

Some errors reported by the interconnect, or poisoned data sent to an interconnect that doesn't support poisoning, will not be reported in the DSU RAS Error Record Registers.

#### Configurations affected

This erratum affects all configurations of the DSU with at least one CHI master port or CHI peripheral port configured.

#### Conditions

1. Either:
  - An uncorrectable ECC error occurs in a core, causing data sent to the DSU to be marked as poisoned.
  - The data is sent to the CHI interconnect, either immediately or later when evicted from the L3.
  - The IMP\_CLUSTERECTLR\_EL1.ENPOISN or IMP\_CLUSTERECTLR\_EL1.ENPOISNPP bits are set to indicate that the interconnect doesn't support poisoned data, or the data is for a DVM operation.
2. Or:
  - An L3 eviction or a DVM message is sent to the CHI interconnect, and the DSU receives an error response.

#### Implications

The original cause of the error happened outside of the DSU and should have been logged by the component that detected the error, although this may have been recorded as a deferred error. The DSU RAS Error Record Registers are not correctly updated to indicate that the deferred error could not be deferred any longer.

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

#### Workaround

No workaround is necessary.

## 1795881

### Trace flush may not flush all data

#### Status

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

#### Description

If a flush request is made on the ATB interface at the same time as a trace source generating trace data is disabled, the flush may complete before the last bytes of the trace data have been output.

#### Configurations affected

This erratum affects configurations of the DSU where the external ATB interface is 64 bits wide.

#### Conditions

1. The ETM in a core or an ELA in a core or the cluster is generating trace output.
2. The ETM or ELA is disabled, therefore it stops generating trace data.
3. The trace subsystem outside the cluster requests a flush by asserting the **AFVALID** signal before the trace data has naturally drained from the cluster.

The cluster may incorrectly assert the **AFREADY** signal before the last trace data has been output on the ATB interface.

#### Implications

Once the trace source has been disabled, the remaining trace data will naturally drain as quickly as it can. If an explicit flush is requested during this time, then the flush may complete before the last bytes of trace data have been output, which could lead to incomplete trace being processed by the debug tools.

#### Workaround

No workaround is necessary.



## 1791722

### ECC errors in L3 Tag RAMs can cause data corruption and loss of coherency

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0

#### Description

Under rare conditions, single and double-bit errors detected by the DSU in the L3 Tag RAMs can lead to data corruption and a loss of coherency.

#### Configurations Affected

This erratum affects all configurations of the DSU with SCU\_CACHE\_PROTECTION enabled and at least one CHI master port.

#### Conditions

1. The interconnect sends a snoop transaction to the DSU, which hits in the L3 cache.
2. The cache line is present in one core in a Unique Dirty state.
3. A core executes an L3 data cache maintenance by set/way instruction.
4. The snoop transaction reads the L3 Tag RAM again and sees a 1-bit or 2-bit ECC error.

If these conditions are met, then the DSU might cause data corruption and a loss of coherency. This might lead to a system deadlock.

#### Implications

If the ERROCTL.ED bit is 1, then the ECC error is recorded in the Error Record Registers, including the ERROSTATUS register. The error is recorded as if the Tag RAM error correction occurred normally and generates any interrupts as normal. As this is an L3 Tag error, unless there is already a more severe error recorded, ERROSTATUS.IERR is set to 0 and ERROSTATUS.SERR is set to 7.

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

#### Workaround

No workaround is necessary.

