



## Arm® DSU-AE (MP092)

### Software Developer Errata Notice

Date of issue: May 03, 2024

Non-Confidential

Document version: 9.0

Copyright © 2018-2024 Arm® Limited (or its affiliates). All rights reserved.

Document ID: SDEN-1343188

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



This document is Non-Confidential.

Copyright © 2018-2024 Arm® Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted Arm's Proprietary notice found at the end of this document.

This document (SDEN\_1343188\_9.0\_en) was issued on May 03, 2024.

There might be a later issue at <http://developer.arm.com/documentation/SDEN-1343188>

## 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).

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® DSU-AE (MP092), create a ticket on <https://support.developer.arm.com>.

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

# Contents

<b>Introduction</b>	5
Scope	5
Categorization of errata	5
<b>Change Control</b>	6
<b>Errata summary table</b>	7
<b>Errata descriptions</b>	8
Category A	8
Category A (rare)	8
Category B	9
1740726 Bit 9 of CLUSTERIFPFAULT<P/R> is asserted incorrectly	9
1750645 Use of FUNC_RET power mode prevents thread wakeup in a multithreaded core	10
Category B (rare)	11
Category C	12
1933388 Interconnect DErr on dirty data not reported in RAS registers	12
2855265 Interconnect bus errors during write back not recorded in RAS registers	14
2231625 CEMODE and CLUSTERCFR safety mode indicator are inconsistent	15
1555811 Incorrect ordering after change in cacheability	17
1314007 The debugger view of the number of processing elements might be incorrect when switching between split and lock modes	19
<b>Proprietary notice</b>	20
<b>Product and document information</b>	22
Product status	22
Product completeness status	22
Product revision status	22

# 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.

## May 03, 2024: Changes in document version v9.0

No new or updated errata in this document version.

## August 04, 2023: Changes in document version v8.0

ID	Status	Area	Category	Summary
<a href="#">2855265</a>	New	Programmer	Category C	Interconnect bus errors during write back not recorded in RAS registers

## March 31, 2022: Changes in document version v7.0

ID	Status	Area	Category	Summary
<a href="#">1750645</a>	New	Programmer	Category B	Use of FUNC_RET power mode prevents thread wakeup in a multithreaded core
<a href="#">1933388</a>	New	Programmer	Category C	Interconnect DErr on dirty data not reported in RAS registers
<a href="#">2231625</a>	Updated	Programmer	Category C	CEMODE and CLUSTERCFR safety mode indicator are inconsistent

## October 28, 2021: Changes in document version v6.0

ID	Status	Area	Category	Summary
<a href="#">2231625</a>	New	Programmer	Category C	CEMODE and CLUSTERCFR safety mode indicator are inconsistent

## October 23, 2020: Changes in document version v5.0

No new or updated errata in this document version.

## May 04, 2020: Changes in document version v4.0

ID	Status	Area	Category	Summary
<a href="#">1740726</a>	New	Programmer	Category B	Bit 9 of CLUSTERIFPFAULT<P/R> is asserted incorrectly

## January 31, 2020: Changes in document version v3.0

ID	Status	Area	Category	Summary
<a href="#">1314007</a>	Updated	Programmer	Category C	The debugger view of the number of processing elements might be incorrect when switching between split and lock modes

## August 26, 2019: Changes in document version v2.0

ID	Status	Area	Category	Summary
<a href="#">1555811</a>	New	Programmer	Category C	Incorrect ordering after change in cacheability

## December 07, 2018: Changes in document version v1.0

ID	Status	Area	Category	Summary
<a href="#">1314007</a>	New	Programmer	Category C	The debugger view of the number of processing elements might be incorrect when switching between split and lock modes

# 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="#">1740726</a>	Programmer	Category B	Bit 9 of CLUSTERIFPFAULT<P/R> is asserted incorrectly	r1p0	r1p1
<a href="#">1750645</a>	Programmer	Category B	Use of FUNC_RET power mode prevents thread wakeup in a multithreaded core	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1933388</a>	Programmer	Category C	Interconnect DErr on dirty data not reported in RAS registers	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2855265</a>	Programmer	Category C	Interconnect bus errors during write back not recorded in RAS registers	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">2231625</a>	Programmer	Category C	CEMODE and CLUSTERCFR safety mode indicator are inconsistent	r0p0, r1p0, r1p1	r1p2
<a href="#">1555811</a>	Programmer	Category C	Incorrect ordering after change in cacheability	r0p0, r1p0, r1p1, r1p2	Open
<a href="#">1314007</a>	Programmer	Category C	The debugger view of the number of processing elements might be incorrect when switching between split and lock modes	r0p0	r1p0

# Errata descriptions

## Category A

There are no errata in this category.

## Category A (rare)

There are no errata in this category.

## Category B

1740726

### Bit 9 of CLUSTERIFPFAULT<P/R> is asserted incorrectly

#### Status

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

#### Description

In the below described configurations, bit 9 of CLUSTERIFPFAULT<P/R> is incorrectly asserted.

#### Configurations Affected

This erratum affects the following configurations of the DSU-AE:

Parameter HYBRID\_MODE set to FALSE

And one of the following CPU configurations:

1. NUM\_LITTLE\_CORES = 6
2. NUM\_LITTLE\_CORES = 8
3. NUM\_LITTLE\_CORES = 4 & NUM\_BIG\_CORES = 2
4. NUM\_LITTLE\_CORES = 4 & NUM\_BIG\_CORES = 4

#### Conditions

1. The SoC asserts bit 9 of CLUSTERIFPCMPEN<P/R>.

#### Implications

When the above conditions are met, bit 9 of CLUSTERIFPFAULT<P/R> will be asserted, potentially masking other faults that would be reported by this bit.

#### Workaround

Do not assert bit 9 of CLUSTERIFPCMPEN<P/R>.



## 1750645

### Use of FUNC\_RET power mode prevents thread wakeup in a multithreaded core

#### Status

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

A core can implement support for the FUNC\_RET and/or the FULL\_RET power modes. When FUNC\_RET is disabled in the CPUPWRCTLR\_EL1.SIMD\_RET\_CTRL field (which is the default), power transitions are allowed directly between the ON and FULL\_RET power modes. When FUNC\_RET is enabled, transitions between FULL\_RET and ON must go through FUNC\_RET, and any direct transitions will be denied as documented in the DSU Technical Reference Manual.

When the DSU is configured with a multithreaded core, the state of the two threads can be controlled with the operating mode in the core P-Channel. If the multithreaded core is in the FULL\_RET power mode, then the core must be moved to the ON power mode before the operating mode can be changed. When used with the Arm Power Policy Unit (PPU) in the PCK-600 product, the PPU will not be aware of the CPUPWRCTLR\_EL1.SIMD\_RET\_CTRL status and will always request a direct transition from FULL\_RET to ON in this situation. The transition will be repeatedly denied by the core, which can lead to a system deadlock.

#### Configurations Affected

This erratum only affects configurations that include a multithreaded core and implement functional retention mode support in the core.

#### Conditions

1. One thread in the core is active.
2. The CPUPWRCTLR\_EL1.SIMD\_RET\_CTRL field is set to a nonzero value.
3. The CPUPWRCTLR\_EL1.WFI\_RET\_CTRL field or the CPUPWRCTLR\_EL1.WFE\_RET\_CTRL field is set to a nonzero value.
4. The active thread executes a WFI or WFE instruction which causes the core to enter FULL\_RET power mode.
5. The PPU receives a wake request for the other thread, and so requests that the core moves to the ON power mode directly from the FULL\_RET mode.

#### Implications

The core will repeatedly deny the transition to ON. If there is no activity that causes the first thread to leave WFE or WFI, then the system will not be able to activate the second thread, which might lead to a system deadlock.

The software should avoid setting the CPUPWRCTLR\_EL1.SIMD\_RET\_CTRL register. If the implementation supported the FUNC\_RET power mode, then this will prevent the benefit of the lower leakage power savings from that mode.

If the implementation wants to support the FUNC\_RET mode, then additional system logic between the cluster and the PPU is possible, please contact Arm for more details about this.

## Category B (rare)

There are no errata in this category.

## Category C

1933388

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

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

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

#### Configurations Affected

This erratum affects configurations of the DSU-AE with at least one CHI master 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. Either of the following occurs:
  - The DSU-AE 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.
  - The interconnect sends a stashing snoop to the DSU-AE and the DSU-AE responds by requesting a Data Pull.
2. The interconnect returns dirty data with a DErr response indicating that there is an error in the data.

#### Implications

The DSU-AE will not allocate the line to L3 because of the error, and so the dirty data will be discarded. The DSU-AE 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-AE 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.

## 2855265

### Interconnect bus errors during write back not recorded in RAS registers

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

When the DSU is writing back data to the system interconnect, errors from the interconnect might not be reported in the DSU RAS Error Record Registers.

#### Configurations Affected

This erratum affects all configurations of the DSU except Direct connect configurations.

#### Conditions

1. The DSU starts a WriteBackFull, WriteCleanFull, or cacheable WriteNoSnpFull transaction to the interconnect. This might be caused by an internally generated transaction, by a CPU transaction, or by an ACP transaction.
2. The interconnect returns an error response, indicating there has been an error during the transaction. For a CHI interconnect, the error response might be NDErr or DErr. For AXI or ACE interconnects, the error response might be SLVERR or DECERR.

#### Implications

The DSU will complete the transaction and will transfer the dirty data to the interconnect. The DSU RAS Error Record Registers are not updated to report the error from the interconnect. If the interconnect might lose the dirty data due to the error, the dirty data might be lost without this being reported in the DSU RAS registers.

The original cause of the error happened outside of the DSU, so this should have been logged by the component that detected the error. Therefore, the system should be able to detect the potential data loss.

#### Workaround

No workaround is necessary.

## 2231625

### CEMODE and CLUSTERCFR safety mode indicator are inconsistent

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1. Fixed in r1p2.

#### Description

The safety mode indicator bits in CLUSTERCFR incorrectly reports the SPLIT mode status as 00 instead of 01.

Configurations Affected

This erratum affects all configurations of the DSU-AE.

#### Conditions

This erratum occurs under the following conditions:

1. The DSU-AE is operating in SPLIT mode (e.g. CEMODE == 01).
2. CLUSTERCFR has been read.
3. The safety mode indicator bits (CLUSTERCFR[31:30]) indicate 00 instead of 01.

#### Implications

Software reading CLUSTERCFR will read safety mode as 00 (RESERVED) instead of 01 (SPLIT), as detailed in the documentation.

The cluster execution mode (CEMODE) is determined by SoC inputs. SPLIT mode is defined as CEMODE[1:0] == 01. The safety mode indicator in the CLUSTERCFR was intended to match CEMODE. Originally, SPLIT mode was defined as 00. SPLIT mode encoding was altered to 01 as result of partner feedback.

When the cluster is operating in LOCK (CEMODE == 11) or HYBRID (CEMODE == 10) the safety mode in CLUSTERCFR is consistent with CEMODE.

Note that SoC explicitly controls the cluster execution mode directly via CEMODE inputs. However, depending on software, reading an incorrect value for safety mode in CLUSTERCFR could potentially have consequences.

#### Workaround

No workaround is necessary for this erratum. The safety mode value read from CLUSTERCFR does not have functional connotations.

## 1555811

### Incorrect ordering after change in cacheability

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2. Open.

#### Description

If the memory type of an address region is changed from Cacheable to Non-cacheable, and then back again, and rare microarchitectural conditions occur, then stale data might be observed in a cache.

#### Configurations Affected

This erratum affects all configurations.

#### Conditions

1. An address region of memory is marked in the translation tables as Write-Back Cacheable memory.
2. The hardware prefetcher starts a data prefetch to an address within this region. This must generate a StashOnce CHI transaction from the core to the DSU-AE, and in some cases the DSU-AE might pass the StashOnce on to the interconnect if the DSU-AE is configured with a CHI master.
3. The translation tables are updated to change the memory type to Non-cacheable or Device memory. This would involve a break-before-make sequence.
4. A sequence of cache clean and invalidate instructions are executed to ensure that any Cacheable data in the memory region does not remain in the caches.
5. The StashOnce transaction and the clean and invalidate transaction to the same address get reordered within the DSU-AE or externally if the StashOnce was sent to the interconnect. This means that the StashOnce transaction can cause the line to be allocated into the cache after the cache maintenance has completed.
6. A core or other master in the system writes to the region that is now marked Non-cacheable or Device.
7. The translation tables are changed a second time, to mark the memory as Write-Back Cacheable again.
8. A load instruction is executed. The load might observe the stale data that was prefetched into the cache, rather than the Non-cacheable data that was written.

#### Implications



The above sequence is very specific and would typically take a very long time to execute. It requires that the StashOnce transaction is started before the translation table modification, yet does not complete until after both the translation table modification and the cache maintenance. Additionally, the StashOnce and cache maintenance transactions must be reordered by the DSU-AE or interconnect, and this is an unlikely event, especially if they are not started at a similar time. Therefore the combination of these conditions is going to be extremely rare. Furthermore, the change in memory type implies a change of use of the memory, and many such changes of use will not require preservation of the data between uses.

## Workaround

No workaround is necessary.

Note that this erratum is caused by a deficiency in the CHI architecture and will be corrected in CHI Issue D onward.

## 1314007

### The debugger view of the number of processing elements might be incorrect when switching between split and lock modes

#### Status

Fault Type: Programmer Category C

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description

The number of entries in the ROM table reflects the number and offsets of the processing elements (PE) in each mode. A cluster reset-time split/lock mode selection might cause the debugger to have an incorrect view of the number of PEs in the current mode.

#### Configurations Affected

This erratum affects all configurations of the DSU-AE.

#### Conditions

1. The SoC dynamically switches between split and lock modes after a Cold reset.
2. The SoC does not reset its debug logic when resetting the cluster and switching modes.
3. The debugger performs debug accesses.

#### Implications

When the above conditions are met, debug accesses might not work as expected because:

- A debugger connected to a part that boots up in lock mode and switches to split mode will not be aware of all the accessible processing elements (PEs).
- A debugger connected to a part that boots up in split mode and switches to lock mode might attempt accessing PEs that are now redundant. Such access will appear to complete but will not have the anticipated effect. Specifically writes will not reach their targets and reads will always return zeros.

#### Workaround

To ensure a debugger has a complete description of the system, the ROM table must be read following each reset of all PEs in the DSU-AE. A reset of a PE can be detected by reading EDPRSR.

# Proprietary notice

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

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

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

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

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

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(PRE-1121-V1.0)

# Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

## Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

### Product completeness status

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

### Product revision status

The [0x0y] identifier indicates the revision status of the product described in this manual, where:

**rx**

Identifies the major revision of the product.

**py**

Identifies the minor revision or modification status of the product.