



# Arm CoreLink CI-700 Coherent Interconnect

## Software Developer Errata Notice

Date of issue: 18-Aug-2023

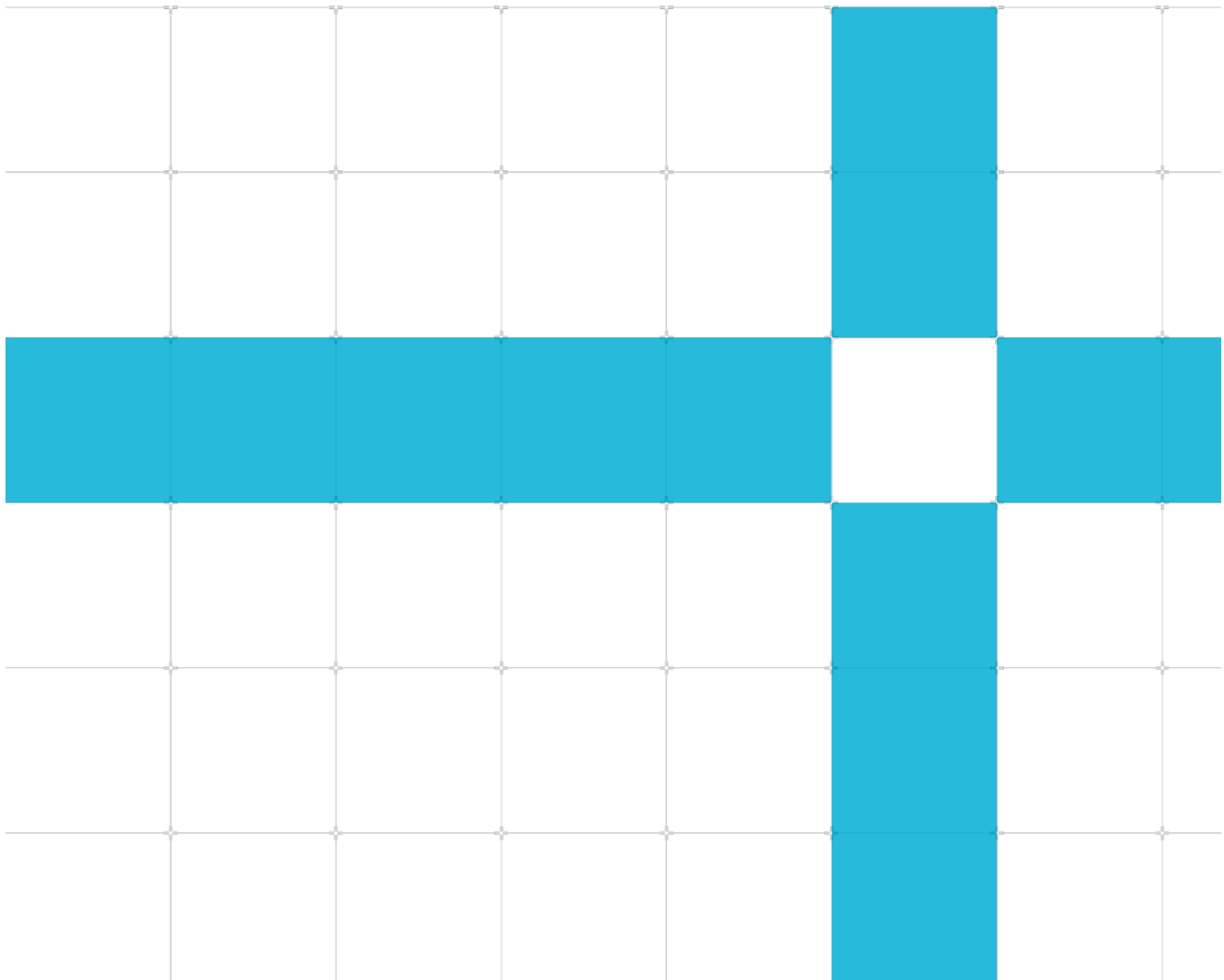
Non-Confidential

Document version: 9.0

Copyright © 2020-2023 Arm® Limited (or its affiliates). All rights reserved.

Document ID: SDEN-1780265

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 © 2020-2023 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 CoreLink CI-700 Coherent Interconnect, 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>	5
Scope	5
Categorization of errata	5
<b>Change Control</b>	6
<b>Errata summary table</b>	8
<b>Errata descriptions</b>	9
Category A	9
Category A (rare)	9
Category B	10
1926789 SECC error on ABF operation can cause coherency failures for other memory addresses	10
2243907 WriteZero transactions can deadlock when not supported downstream of MTSX	12
2301815 MTE Tag Cache transaction queue may hang	13
3013639 Write Stash can cause multi-copy atomicity issue	14
Category B (rare)	15
Category C	16
2177971 HN-I RAS syndrome registers do not capture correct opcode	16
2741290 RAS HN-I and SBSX ERRGSR registers do not capture correct device instance information	18
3013642 Incorrect TagMatch response on partial writes with MTE Match	19
3031700 Debug reads with simultaneous coherent traffic or dynamic power transitions can cause deadlock	20

# 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.

## 18-Aug-2023: Changes in document version v9.0

ID	Status	Area	Category	Summary
<a href="#">3013639</a>	New	Programmer	Category B	Write Stash can cause multi-copy atomicity issue
<a href="#">3013642</a>	New	Programmer	Category C	Incorrect TagMatch response on partial writes with MTE Match
<a href="#">3031700</a>	New	Programmer	Category C	Debug reads with simultaneous coherent traffic or dynamic power transitions can cause deadlock

## 21-Sep-2022: Changes in document version v8.0

ID	Status	Area	Category	Summary
<a href="#">2741290</a>	New	Programmer	Category C	RAS HN-I and SBSX ERRGSR registers do not capture correct device instance information

## 11-Apr-2022: Changes in document version v7.0

No new or updated errata in this document version.

## 11-Feb-2022: Changes in document version v6.0

ID	Status	Area	Category	Summary
<a href="#">2301815</a>	Updated	Programmer	Category B	MTE Tag Cache writes may be dropped

## 07-Jan-2022: Changes in document version v5.0

ID	Status	Area	Category	Summary
<a href="#">2301815</a>	New	Programmer	Category B	MTE Tag Cache writes may be dropped

## 27-Sep-2021: Changes in document version v4.0

ID	Status	Area	Category	Summary
<a href="#">2243907</a>	New	Programmer	Category B	WriteZero transactions can deadlock when not supported downstream of MTSX
<a href="#">2177971</a>	New	Programmer	Category C	HN-I RAS syndrome registers do not capture correct opcode

## 25-Sep-2020: Changes in document version v3.0

No new or updated errata in this document version.

## 21-Aug-2020: Changes in document version v2.0

ID	Status	Area	Category	Summary
<a href="#">1926789</a>	New	Programmer	Category B	SECC error on ABF operation can cause coherency failures for other memory addresses

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

No errata in this document version.

# 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="#">1926789</a>	Programmer	Category B	SECC error on ABF operation can cause coherency failures for other memory addresses	r0p0	r1p0
<a href="#">2243907</a>	Programmer	Category B	WriteZero transactions can deadlock when not supported downstream of MTSX	r0p0, r1p0, r1p1, r2p0	r1p2, r2p1
<a href="#">2301815</a>	Programmer	Category B	MTE Tag Cache writes may be dropped	r0p0, r1p0, r1p1, r1p2, r2p0	r1p3, r2p1
<a href="#">3013639</a>	Programmer	Category B	Write Stash can cause multi-copy atomicity issue	r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1	Open
<a href="#">2177971</a>	Programmer	Category C	HN-I RAS syndrome registers do not capture correct opcode	r0p0, r1p0, r1p1, r1p2, r1p3	r2p0
<a href="#">2741290</a>	Programmer	Category C	RAS HN-I and SBSX ERRGSR registers do not capture correct device instance information	r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1	Open
<a href="#">3013642</a>	Programmer	Category C	Incorrect TagMatch response on partial writes with MTE Match	r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1	Open
<a href="#">3031700</a>	Programmer	Category C	Debug reads with simultaneous coherent traffic or dynamic power transitions can cause deadlock	r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1	Open



# Errata descriptions

## Category A

There are no errata in this category.

## Category A (rare)

There are no errata in this category.

## Category B

1926789

### SECC error on ABF operation can cause coherency failures for other memory addresses

#### Status:

Affects: CI-700

Fault Type: Programmer Category B

Fault Status: Present in r0p0. Fixed in r1p0.

#### Description:

CI-700 supports Address Based Flush (ABF) where upper and lower system addresses can be programmed and then request a hardware-based engine to flush out that address range from all System Level Caches (SLC). This ABF state machine works in the presence of other memory requests.

Single-bit ECC errors on the ABF accesses can corrupt the CMN Snoop Filter state, and result in coherency failures for other unrelated memory addresses.

#### Configurations Affected:

Any configuration of CI-700 where ABF is used.

#### Conditions:

This bug appears when following three conditions occur:

- SLC address from flush set/way is outside ABF programmed range AND
- SLC Tag read has single bit ECC error AND
- There is independent request in pipeline N cycles ahead of ABF request (where N is SLC\_TAG\_RAM\_LATENCY)  
In this case, ABF request corrupts SF vector for independent request that's ahead of ABF causing coherency failure.

#### Implications:

The ABF flush sequence can cause coherency fails for unrelated memory addresses during the sequence.

#### WorkAround:

Use the CI-700 power management features to flush the SLC, flushes the full SLC contents vs. the upper/lower range.

## 2243907

# WriteZero transactions can deadlock when not supported downstream of MTSX

## Status

Affects: CI-700

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r2p0. Fixed in r1p2, r2p1.

## Description

If AXI WriteZero transactions are not supported downstream of MTSX, CI-700 can be configured to synthesize WriteZero data. Interactions between sequences of WriteZero transactions that synthesize data can cause write transactions to deadlock.

## Configurations affected

CI-700 configurations where MTSX AXI WriteZero propagation is not enabled and RN-Fs issue WriteZero CHI transactions.

## Conditions

- MTE is enabled AND
- MTSX is configured to synthesize WriteZero data AND
- Sequences of WriteZero transactions where the writes are issued to AXI and internal MTE logic out of order

## Implications

If the preceding conditions are met, WriteZero transactions can deadlock.

## Workaround

Disable WriteZero in the RN-F. Arm CPUs support WriteZero disable. For example:

- Cortex-A710 core, CPUACTLR5\_EL1.l2spr\_writezero\_dis[29] = 1
- Cortex-X2 core, CPUACTLR5\_EL1.l2spr\_writezero\_dis[29] = 1
- Cortex-A510, IMP\_CPUECTLR\_EL1.wzdis[18] = 1

Note that disabling WriteZero in Arm CPUs may reduce WriteZero performance.

## 2301815

### MTE Tag Cache transaction queue may hang

#### Status

Affects: CI-700

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r1p0, r1p1, r1p2, r2p0. Fixed in r1p3, r2p1.

#### Description

Transactions with MTE enabled that share the same tag DRAM address can lead to a hang in the transaction tracking queue.

#### Configurations affected

CI-700 configurations with MTSX configured with a Tag Cache.

#### Conditions

- MTE is enabled AND
- 2 MTE Tag requests to the same 32\*64B memory region AND
- MTE Tag Cache enabled AND
- Sufficient traffic leading to a Tag Cache eviction with matching tag DRAM address

#### Implications

Either of the following:

- Hang: the MTU transaction tracking queue is hung and no forward progress can occur. A reset is required.
- Tag data corruption: the youngest MTE Tag data can be overwritten by the next youngest write, one of the write data is dropped.

#### Workaround

Disable the MTSX MTE Tag Cache.

## 3013639

### Write Stash can cause multi-copy atomicity issue

#### Status

Affects: CI-700

Fault Type: Programmer Category B

Fault Status: r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1. Open.

#### Description

CHI and AXI Write Stash operations can incorrectly get early completion before snooping is complete causing multi-copy atomicity issues.

For example, an RN-I or RN-D PCI MSI write issued after a Write Stash can result in the CPU having an older or stale copy of the Write Stash data at the time of the MSI interrupt.

Another example is an RN-I or RN-D write flag issued after completion of the Write Stash, the CPU can observe the flag update before the Write Stash data is updated.

Note that Arm CPUs do not issue Write Stash transactions.

#### Configurations affected

Any CMN configuration.

#### Conditions

This erratum occurs when the following conditions are met:

- RN-I or RN-D issues AXI Write Stash transaction with a valid StashNID targeting a CPU cache
- RN-I or RN-D issues another AXI transaction after receiving the completion for the Write Stash. For example, PCIE MSI write or write to flag address
- The Stash CPU can observe the results of the second transaction above before the Write Stash data is updated for the first

#### Implications

If the conditions are met, Write Stash might receive early completion while the Stash CPU still has an old copy causing multi-copy atomicity issues.

#### Workarounds

The workaround is to send the result in Stash to the SLC instead of the CPU cache, by disabling stash snooping using `por_hnf_aux_ctl.hnf_stash_disable`.

## Category B (rare)

There are no errata in this category.

## Category C

### 2177971

### HN-I RAS syndrome registers do not capture correct opcode

#### Status

Affects: CI-700

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r1p0, r1p1, r1p2, r1p3. Fixed in r2p0.

#### Description

The OPCODE field in the HN-I por\_hni\_errmisc RAS Syndrome register does not correctly capture the new REQ opcodes introduced in CHI-E.

#### Configurations Affected

All CI-700 configurations that use RAS error logging.

#### Conditions

A RAS error triggered by a new CHI-E transaction that causes the syndrome to be captured in the por\_hni\_errmisc register on a transaction processed by HN-I/P/D/V/T.

#### Implications

A read of the por\_hni\_errmisc.OPCODE field may return an incorrect opcode. The opcode does not properly reflect an error on a CHI-E opcode that has bit [6] set.

#### Workaround

RAS handler and software can use the following table indicating which por\_hni\_errmisc.OPCODE values are affected by aliasing due to this issue. If a RAS error involves opcodes listed as **Yes**, software can indicate that either opcode could have been the actual opcode involved in the error. Note that some cases with opcode[6]=0 are Reserved in the *CHI-E Specification*.



CHI-E REQ Opcodes			
Opcode[5:0]	Opcode[6]=0	Opcode[6]=1	Can Opcode[6]=1 RAS error happen at HN-X?
0x01	ReadShared	MakeReadUnique	Yes
0x02	ReadClean	WriteEvictOrEvict	No
0x03	ReadOnce	WriteUniqueZero	Yes
0x04	ReadNoSnp	WriteNoSnpZero	No
0x07	ReadUnique	StashOnceSepShared	No
0x08	CleanShared	StashOnceSepUnique	No
0x0C	MakeUnique	ReadPreferUnique	Yes
0x10	Reserved	WriteNoSnpFullCleanSh	No
0x11	ReadNoSnpSep	WriteNoSnpFullCleanInv	No
0x12	Reserved	WriteNoSnpFullCleanSh-PerSep	No
0x14	DVMOp	WriteUniqueFullCleanSh	Yes
0x16	Reserved (WriteCleanPtl)	WriteUniqueFullCleanSh-PerSep	Yes
0x18	WriteUniquePtl	WriteBackFullCleanSh	Yes
0x19	WriteUniqueFull	WriteBackFullCleanInv	Yes
0x1A	WriteBackPtl	WriteBackFullCleanSh-PerSep	Yes
0x1C	WriteNoSnpPtl	WriteCleanFullCleanSh	Yes
0x1E	Reserved	WriteCleanFullCleanSh-PerSep	Yes
0x20	WriteUniqueFullStash	WriteNoSnpPtlCleanSh	No
0x21	WriteUniquePtlStash	WriteNoSnpPtlCleanInv	No
0x22	StashOnceShared	WriteNoSnpPtlCleanSh-PerSep	No
0x24	ReadOnceCleanInvalid	WriteUniquePtlCleanSh	Yes
0x26	ReadNotSharedDirty	WriteUniquePtlCleanSh-PerSep	Yes

## 2741290

### RAS HN-I and SBSX ERRGSR registers do not capture correct device instance information

#### Status

Affects: CI-700

Fault Type: Programmer Category C

Fault Status: r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1. Open.

#### Description

The CI Error Group Status Registers (ERRGSR) capture device instance error information for RAS events. The registers indicate the device instance within a device group. The registers are not updated correctly for the HN-I and SBSX device groups, so cannot be used to determine the device instances for RAS events.

#### Configurations Affected

All CI-700 configurations that use RAS error logging.

#### Conditions

A RAS event triggered by an HN-I or SBSX device.

#### Implications

Software cannot use the HN-I or SBSX ERRGSR registers.

#### Workaround

The RAS handler must read the individual HN-I and SBSX instance RAS logging registers when RAS interrupts occur.

## 3013642

### Incorrect TagMatch response on partial writes with MTE Match

#### Status

Affects: CI-700

Fault Type: Programmer Category C

Fault Status: r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1. Open.

#### Description

Partial Write requests with MTE TagOp Match can cause an incorrect TagMatch response

#### Configurations affected

Any configuration with HN-F devices that use MTE without MTSX

#### Conditions

This erratum occurs when the following conditions are met:

- Non-Arm CPU issues non-allocating WriteUniquePtl with TagOp=Match and Tag=<partial>
- The System Level Cache has dirty data but without MTE Tag
- HN-F incorrectly responds with no TagMatch for the WriteUniquePtl

#### Implications

If the conditions are met, MTE Write Partial transactions that require TagMatch response can be incorrect. Partial write transactions might not respond with TagMatch.

#### Workarounds

No workaround required for Arm CPUs or configurations with MTSX, Arm CPUs do not issue Write Partial with TagMatch.

## 3031700

### Debug reads with simultaneous coherent traffic or dynamic power transitions can cause deadlock

#### Status

Affects: CI-700

Fault Type: Programmer Category C

Fault Status: r0p0, r1p0, r1p1, r1p2, r1p3, r2p0, r2p1, r2p2, r3p0, r3p1. Open.

#### Description

HN-F System Level Caches (SLC) and Snoop Filter (SF) Debug Reads with simultaneous coherent traffic or dynamic power retention transitions can cause a deadlock.

#### Configurations affected

Any configuration.

#### Conditions

This erratum occurs when one of the following conditions are met:

- Coherent transactions that require HN-F Snoop Filter allocation while performing SLC or SF debug read
- Dynamic retention mode is enabled while performing a SLC or SF debug read

#### Implications

A deadlock can occur if the conditions are met. Note that expected usage is performing the Debug Reads in the absence of traffic since traffic can change the state of the RAMs.

#### Workaround

Use the following workarounds to prevent a deadlock:

- Stop CPU (RN-F) and IO (RN-I) coherent traffic before issuing Debug Reads
- Disable Dynamic retention power transitions via `por_hnf_ppu_pwpr.dyn_en = 1'b0` (reset value)

