CoreSight™ SoC-400 (TM100)

Software Developer Errata Notice

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

Non-Confidential Proprietary notice

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.

Web address

http://www.arm.com/.

Feedback on this product

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

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

Feedback on this document

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

- The document title.
- The document number: SDEN-1357391.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Contents

Introduction

Scope

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

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

Categorization of errata

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

Category A A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.

Category A (Rare) A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.

Category B A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.

Category B (Rare) A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.

Category C A minor error.

Change control

Errata are listed in this section if they are new to the document, or marked as “updated” if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The errata summary table on page 9 identifies errata that have been fixed in each product revision.

15-Nov-2019: Changes in document version 12.0				
ID	Status	Area	Cat	Summary of erratum
1624041	New	Programmer	CatC	AHB-AP can issue transactions where HADDR[1:0] is not aligned to HSIZE on the AHB

30-Apr-2019: Changes in document version 11.0				
ID	Status	Area	Cat	Summary of erratum
585521	Updated	Programmer	CatB	(854072) Timestamp decoder resynchronisation livelock
568693	Updated	Programmer	CatB	(854071) Resync packet on slave causes missing packet on master

27-Feb-2019: Changes in document version 10.0				
ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

24-Oct-2016: Changes in document version 9.0				
ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

29-Jan-2016: Changes in document version 8.0				
ID	Status	Area	Cat	Summary of erratum
585521	New	Programmer	CatB	(854072) Timestamp decoder resynchronisation livelock
568693	New	Programmer	CatB	(854071) Resync packet on slave causes missing packet on master
457091	Updated	Programmer	CatB	(841819) ATB async bridge loses last flush data

11-Mar-2015: Changes in document version 7.0				
ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

10-Mar-2015: Changes in document version 6.0				
ID	Status	Area	Cat	Summary of erratum

457091	New	Programmer	CatB	(841819) ATB async bridge loses last flush data
408543	Updated	Programmer	CatA (rare)	(832019) Timestamp replicator can stall synchronisation
337153	Updated	Programmer	CatB	(806419) Timestamps output by the timestamp decoder can be incorrect after synchronizing to a running timestamp stream

24-Jul-2014: Changes in document version 5.0

ID	Status	Area	Cat	Summary of erratum
408543	New	Programmer	CatA (rare)	(832019) Timestamp replicator can stall synchronisation
337153	Updated	Programmer	CatB	(806419) Timestamps output by the timestamp decoder can be incorrect after synchronizing to a running timestamp stream

18-Jun-2014: Changes in document version 4.0

ID	Status	Area	Cat	Summary of erratum
399409	New	Programmer	CatC	(829169) ATB async bridge can generate spurious trace output after exit from low power state
358129	New	Programmer	CatA (rare)	(820419) Narrow Timestamp can be corrupted when generated timestamp is stopped

15-Nov-2013: Changes in document version 3.0

ID	Status	Area	Cat	Summary of erratum
341182	New	Programmer	CatC	(813569) CSTPIU fails to output sync after the pattern generator is disabled in Normal mode
337157	New	Programmer	CatC	(798174) Narrow timestamp synchronous bridge might reduce timestamp resolution
337156	New	Programmer	CatB	(771721) When accessing AHB-AP BD0-BD3 registers, HADDR[1:0] might not be aligned to HSIZE on the AHB
334938	New	Programmer	CatB	(814223) SWJDP sets parity error in Overrun mode

02-Oct-2013: Changes in document version 2.0

ID	Status	Area	Cat	Summary of erratum
328341	New	Programmer	CatC	(771723) CoreSight SW-DP ignores data parity errors in writes to the ABORT register
323937	New	Programmer	CatB	(811981) Narrow timestamp synchronization can be delayed

01-Jul-2013: Changes in document version 1.0				
ID	Status	Area	Cat	Summary of erratum
337154	New	Programmer	CatB	(774720) DbgSwEnable should be HIGH when DeviceEn is LOW
337153	New	Programmer	CatB	(806419) Timestamps output by the timestamp decoder can be incorrect after synchronizing to a running timestamp stream
337151	New	Programmer	CatB	(806420) AXI-AP Debug Base Address Register returns upper and lower 32 bits swapped
275600	New	Programmer	CatC	(801369) APB-AP supports the ROM table only at 0x80000000
260244	New	Programmer	CatC	(794419) Timestamp Encoder stops synchronizing the timestamp when the input counter stops counting
209972	New	Programmer	CatB	(771719) CTIINTACK register needs clearing each time it is set
209589	New	Programmer	CatC	(771724) CTI Authentication Status register is incorrect
202988	New	Programmer	CatC	(773919) The designer of AXI-AP is incorrectly identified by the identification register

Errata summary table

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

ID	Cat	Summary	Found in versions	Fixed in version
1624041	CatC	AHB-AP can issue transactions where HADDR[1:0] is not aligned to HSIZE on the AHB	R0P0, R1P0, R2P0, R2P1, R3P0, R3P1, R3P2, R3P2-50	
585521	CatB	(854072) Timestamp decoder resynchronisation livelock	R3P2	
568693	CatB	(854071) Resync packet on slave causes missing packet on master	R3P2	
457091	CatB	(841819) ATB async bridge loses last flush data	R0P0, R1P0, R2P0, R2P1, R3P0, R3P1	R3P2
408543	CatA (rare)	(832019) Timestamp replicator can stall synchronisation	R0P0, R1P0, R2P0, R2P1, R3P0, R3P1	R3P2
399409	CatC	(829169) ATB async bridge can generate spurious trace output after exit from low power state	R0P0, R1P0, R2P0, R2P1, R3P0, R3P1	R3P2
358129	CatA (rare)	(820419) Narrow Timestamp can be corrupted when generated timestamp is stopped	R3P0, R3P1	R3P2
341182	CatC	(813569) CSTPIU fails to output sync after the pattern generator is disabled in Normal mode	R0P0, R1P0, R2P0, R2P1, R3P0, R3P1, R3P2	
337157	CatC	(798174) Narrow timestamp synchronous bridge might reduce timestamp resolution	R0P0, R1P0	R2P0
337156	CatB	(771721) When accessing AHB-AP BD0-BD3 registers, HADDR[1:0] might not be aligned to HSIZE on the AHB	R0P0	R1P0
337154	CatB	(774720) DbgSwEnable should be HIGH when DeviceEn is LOW	R0P0	R1P0
337153	CatB	(806419) Timestamps output by the timestamp decoder can be incorrect after synchronizing to a running timestamp stream	R0P0, R1P0, R2P0, R2P1, R3P0	R3P1
337151	CatB	(806420) AXI-AP Debug Base Address Register returns upper and lower 32 bits swapped	R0P0, R1P0, R2P0, R2P1	R3P0
334938	CatB	(814223) SWJDP sets parity error in Overrun mode	R1P0, R2P0, R2P1, R3P0, R3P1	R3P2
328341	CatC	(771723) CoreSight SW-DP ignores data parity errors in writes to the ABORT register	R0P0	R1P0
323937	CatB	(811981) Narrow timestamp synchronization can be delayed	R0P0, R1P0, R2P0, R2P1, R3P0	R3P1
275600	CatC	(801369) APB-AP supports the ROM table only at 0x80000000	R0P0, R1P0, R2P0, R2P1	R3P0
260244	CatC	(794419) Timestamp Encoder stops synchronizing the timestamp when the input counter stops counting	R0P0, R1P0	R2P0
209972	CatB	(771719) CTIINTACK register needs clearing each time it is set	R0P0	R1P0

ID	Cat	Summary	Found in versions	Fixed in version
209589	CatC	(771724) CTI Authentication Status register is incorrect	R0P0	R1P0
202988	CatC	(773919) The designer of AXI-AP is incorrectly identified by the identification register	R0P0	R1P0

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

408543

(832019) Timestamp replicator can stall synchronisation

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category A (rare)
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0, r3p1, Fixed in r3p2

Description
This erratum affects the following components:

- Narrow timestamp replicator.
    - cxntsreplicator
    - Component Revisions: r0p0, r1p0

The timestamp replicator is used to connect several timestamp consumers to a single timestamp source. The replicator operates to ensure that all consumers are able to synchronise at any time by adapting the speed of the synchronization channel. Higher resolution timestamp updates are also provided to all timestamp consumers to use once synchronization has been achieved. As a result of this erratum, specific clock ratio conditions can result in the synchronization channel becoming stalled.

Conditions

- The narrow timestamp replicator is used in a timestamp network.
- At least two of the timestamp replicator master ports are connected to narrow timestamp bridges, where the bridges operate with the slave port clocked faster than the master port.
- The clock ratios of the timestamp bridges are correlated, resulting in a condition where TSSYNCREADYM for all of the ports of the replicator do not become HIGH on the same cycle (even after several cycles of TSSYNCREADYM on each port)

Implications
The condition is only likely to occur when the timestamp consumers are in an active low power condition. Even with all of the consumers running at the same clock ratio to the timestamp replicator, it is likely that the bridges will be out of step but all deriving a clock from the same source.

In this condition, if the timestamp decoder loses synchronization (as the result of a reset of any part of the timestamp interconnect, or an update to the source timestamp) then the decoder will not synchronise until the clock ratios change. If the decoder is not synchronized, it will generate a timestamp value of zero (which is interpreted as indicating that no valid timestamp is available). When used for CoreSight timestamp distribution, this will prevent correlation of trace sources but has no direct impact on the operation of most trace sources.

If used for CPU time, this will prevent a processor from booting.

Workaround
Any workaround will be specific to a system timestamp topology. If it is possible to observe the decoded timestamp, some systems will be able to recover from this condition by adjusting the system clock ratios for long enough to permit all of the timestamp decoders in a system to achieve synchronization. This requires typically 128 cycles of the slowest clock between timestamp generator and timestamp decoder.

358129

(820419) Narrow Timestamp can be corrupted when generated timestamp is stopped

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category A (rare)
Fault Status: Present in: r3p0, r3p1, Fixed in r3p2

Description
This erratum affects the following components:

- Narrow timestamp asynchronous bridge.
    - cxntsasyncbridge
    - Component Revisions: r0p1, r0p2
- Narrow timestamp synchronous bridge.
    - cxntssyncbridge
    - Component Revisions: r0p2, r0p3

The narrow timestamp bridges support operation where the destination (master interface) is clocked slower than the source (slave interface). In this scenario, timestamp values are propagated but with reduced resolution. As a result of this erratum, if the source timestamp value does not change value on every cycle (or the bridge is driven by a timestamp replicator) then it is possible for the decoded timestamp value to jump backwards. In the worst case, the timestamp value will not converge to the correct value until the timestamp has doubled in value.

Conditions

- The narrow timestamp bridge is used in a configuration where the FIFO can become full. If the FIFO is 6 entries deep, this requires clkm to be slower than clks.
- The tsbits input to the bridge is driven with 0. This requires one of the following to occur:
    - The input value to the encoder to be static, for example, if the timestamp generator is stopped.
    - The bridge is driven by the narrow timestamp replicator, where one or more of the other outputs from the replicator is routed to a bridge, which crosses to a slower clock domain.
    - The timestamp source for the bridge has already been bridged from a slow clock domain to a faster clock domain.

The timestamp input value will normally only be static if it is being used for system time and the system enters debug state.

Implications
If the bridge is being used to distribute generic system time and if the timestamp jumps backwards, then this can have adverse effects on timers and other software or hardware dependent on the time value.
If the bridge is being used to distribute CoreSight timestamps and if the timestamp jumps backwards, this can mean that the debug tools using timestamps to correlate trace streams might perform incorrect correlation.

If the system is affected by this erratum, there is no way to determine that the timestamp value has been corrupted.

Workaround
There is no software workaround. The erratum can be avoided by using the r2p1 revision of the narrow timestamp synchronous or asynchronous bridge in conjunction with r3p1 revisions of the other timestamp distribution components.

Category B

585521

(854072) Timestamp decoder resynchronisation livelock

Status
Affects: CoreSight SoC-400 - Perpetual
Fault type: Programmer Category B
Fault status: Present in r3p2, Fixed in Open

Description

This erratum affects the following components:

- Narrow timestamp asynchronous bridge.
    - cxntsasyncbridge
    - Component Revisions: r1p0

The narrow timestamp asynchronous bridge supports operation where the destination (master interface) is clocked at a lower or equal frequency to the source (slave interface). The bridge contains a feature to detect the master clock being stopped or running much slower than the slave clock, which issues a resync token on the master port when this feature is triggered. In normal operation it is possible for the narrow timestamp value, tsbits[6:0] to have the same value in consecutive cycles, which indicates that the timestamp hasn't changed. As a result of this erratum, the clock stop detection logic may be falsely triggered by the timestamp value on the input taking the same value as in the previous clock cycle.

Conditions

- The narrow timestamp async bridge is used in a configuration where the FIFO can become full. If the FIFO is 4 or 6 entries deep, this requires clkm to be equal to or slower than clks.
- The nts async bridge is preceded by nts sync bridge or nts async bridge.

Implications

As a result of this erratum, the ntsasyncbridge erroneously triggers the slow clock detection feature. On receipt of the resync token, the downstream decoders go into synchronization mode. Before the synchronization is finished, the slow clock feature may trigger again, which causes the decoder to restart synchronization, and so on, leading to a livelock, with decoder always in synchronization mode and outputting a zero decoded timestamp.

When in synchronization mode, the decoder outputs zero on the wide timestamp interface. So any components using timestamp value will receive an invalid timestamp value of zero.

Workaround

These workarounds are for system implementers. There are two possible workarounds
1) The erratum can be avoided by using the r2p1 revision of the narrow timestamp asynchronous bridge in conjunction with r3p2 revisions of the other timestamp distribution components.
2) Set the THRESHOLD value to 127 in the RTL. This needs to be done in the rendered area, as the maximum, value allowed by IP-XACT for THREHOLD in r3p2 is 64. When set to 127, the slow clock detection feature will be disabled.

568693

(854071) Resync packet on slave causes missing packet on master

Status
Affect: CoreSight SoC-400 - Perpetual
Fault type: Programmer Category B
Fault status: Present in r3p2, Fixed in Open

Description
This erratum affects the following components:

- Narrow timestamp asynchronous bridge.
  - cxntsasyncbridge
  - Component Revisions: r1p0

The narrow timestamp bridge supports operation where the destination (master interface) is clocked at a lower or equal frequency to the source (slave interface). The bridge contains a feature to detect the master clock being stopped or running much slower than the slave clock, which issues a resync token on the master port when this feature is triggered. In normal operation it is possible for the narrow timestamp value, tsbits[6:0] to have the same value in consecutive cycles, which indicates that the timestamp hasn't changed. As a result of this erratum, the clock stop detection logic may be falsely triggered by a RESYNC packet (7'd127) received on the slave port, tsbits[6:0].

Conditions

- The narrow timestamp async bridge is used in a configuration where the FIFO can become full. If the FIFO is 4 or 6 entries deep, this requires clkm to be equal to or slower than clks.
- The narrow timestamp async bridge affected by this erratum is preceded by narrow timestamp sync bridge or async bridge.
- The narrow timestamp async bridge affected by this erratum receives a RESYNC packet on the slave port tsbits[6:0]

Implications
As a result of this erratum, the ntsasyncbridge erroneously triggers the slow clock detection feature on receipt of the RESYNC token on the slave port tsbits[6:0]. As a result of this the decoder synchronizes to an incorrect timestamp value.

Workaround
These workarounds are for system implementers. There are two possible workarounds
1) The erratum can be avoided by using the r2p1 revision of the narrow timestamp asynchronous bridge in conjunction with r3p2 revisions of the other timestamp distribution components.
2) Set the THRESHOLD value to 7'd127 in the RTL. This needs to be done in the rendered area, as the maximum, value allowed by IP-XACT for THRESHOLD in r3p2 is 64. When set to 7'd127, the slow clock detection feature will be disabled.

457091

(841819) ATB async bridge loses last flush data

Status
Affects: CoreSight SoC-400 - Perpetual
Fault type: Programmer Category B
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0, r3p1, Fixed in r3p2

Description
This erratum affects the following components:

- ATB asynchronous bridge.
  - cxatbasyncbridge
  - Component Revisions: r0p0, r0p1

When the ATB asynchronous bridge (slave side) receives a flush request or is powered down using the LPI interface it requests a flush of the upstream trace components. At the end of the flush the trace source connected to the slave interface informs the bridge that flush is complete using the afreadys signal. When this is received the bridge should pass all trace data supplied up to and including that point to the master port and signal completion to the downstream component using afreadym. With this erratum it is possible for the last trace transaction to be lost or delayed within the bridge. The last transaction to be output at the end of the flush will be the penultimate transaction received on the slave interface when flush completion is signalled.
1) In the case of an LPI triggered flush, the last transaction is lost permanently.
2) In the case of a normal flush requested by a downstream component, the last transaction is delayed in the bridge and output at a later time, such as during another flush or drains out in normal operation.

Conditions
1) The ATB asynchronous bridge is used in the trace network.
2) A flush request is received or the LPI is used to power down the slave side of the bridge, causing a flush request of upstream trace components.
3) The ATB asynchronous bridge is being used to bridge between two dissimilar clock domains (if the clocks are synchronous then the erratum will not be seen).

Implications
Criticality of the loss of the final transaction being flushed is user scenario dependent.
Losing the last transaction might give rise to synchronisation issues because it might not be possible to decompress the next part of the stream. However, most trace streams will perform trace stream synchronization after a power-down and therefore the impact of this erratum is likely to be limited.
In the case where a flush has resulted in the last transaction being delayed inside the bridge, performing a second flush will ensure that the data is output from the bridge.

Workaround
There is no means to prevent the final transaction from being lost.
Users should ensure that enough trace output is generated that the final transaction is not important.

337156

(771721) When accessing AHB-AP BD0-BD3 registers, HADDR[1:0] might not be aligned to HSIZE on the AHB

Status

Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r0p0 Fixed in r1p0.

Description

This erratum affects the following components:

- AHB Access Port.
  - cxdapahbap
  - Component Revisions: r0p5

The ARM Debug Interface v5 Architecture Specification specifies a TAR (Transfer Address Register) in the MEM-AP that holds the memory address to be accessed. The MEM-AP also has Banked Data Registers 0-3 (BD0-BD3). The addresses of BD0-BD3 are derived from the TAR as mentioned in ADIv5 specification. TAR[3:0] must be ignored in this address calculation and HADDR[1:0] must be forced to 2'b00.
However, the AHB-AP does not force HADDR[1:0] to 2'b00 when BD0-BD3 are accessed. TAR[1:0] is used to drive HADDR[1:0] when BD0-BD3 are accessed. This can result in illegal AHB transactions that are not correctly aligned according to HSIZE if HADDR[1:0] is programmed with an unaligned value.

Conditions

1) TAR[1:0] programmed with a value that is not aligned with the size programmed in the CSW register of the AHB-AP.
2) Initiate an access to a Banked Data Register (BD0-BD3) in the AHB-AP.

Implications

As a result of the programming conditions listed above, AHB-AP erroneously initiates an access on the AHB with HADDR[1:0] not aligned to the size on HSIZE. This might initiate an illegal AHB access. This behavior is not a problem for users using the AHB-AP in a normal usage pattern in which the programmed address is aligned to the size.

Workaround

This is a workaround for users and tools vendors. Before accessing BD0-BD3 registers, ensure that TAR[1:0] is programmed with a value that is aligned to the size value programmed in the CSW register.

337154

(774720) DbgSwEnable should be HIGH when DeviceEn is LOW

Status

Affects: CoreSight SoC - Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r0p0 Fixed in r1p0.

Description

This erratum affects the following components:

- APB Access Port.
    - cxdapapbap
    - Component Revisions: r0p3

It is expected that self-hosted debug must not be disabled when the external debug is disabled. In the APB-AP, DbgSwEnable bit of the APB Control/Status register can be set to low even when DEVICEEN input is low. The PDBGSWEN output takes the value of DbgSwEnable bit and controls software accesses to debug components

Conditions

1) Drive DEVICEEN input low
2) Write 0 to the DbgSwEnable bit of APB Control/Status register On-chip software is accessing one of the following:

- Debug-APB components
- CP14 debug registers in a processor.

Implications

This erratum may lead to disabling software access to the debug components when DEVICEEN is low.
The ARM Debug Interface recommends that DEVICEEN is normally tied high.

Workaround

This is a workaround for system implementers.
If DEVICEEN can take any value other than HIGH, glue logic must be implemented outside the APB-AP to create a signal:
PDBGSWEN_new = ~DEVICEEN || PDBGSWEN
PDBGSWEN_new can then be used to connect to all destinations of PDBGSWEN, including the CoreSight SoC APB interconnect

337153

(806419) Timestamps output by the timestamp decoder can be incorrect after synchronizing to a running timestamp stream

Status

Affects: CoreSight SoC-400- Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0 Fixed in r3p1.

Description

This erratum affects the following components:

- Timestamp decoder.
  - cxtsd
  - Component Revisions: r0p0, r0p1
- Narrow timestamp asynchronous bridge.
  - cxntsasyncbridge
  - Component Revisions: r0p0, r0p1
- Narrow timestamp synchronous bridge.
  - cxntssyncbridge
  - Component Revisions: r0p0, r0p1
- Narrow timestamp replicator.
  - cxntsreplicator
  - Component Revisions: r0p0

The timestamp decoder component decodes the compressed timestamp stream into a full 64-bit timestamp. The compressed timestamp stream consists of the following channels:
• Synchronization channel: Gives the current timestamp value.
• Binary count channel: Gives timestamp information relative to previous timestamps.
The timestamp decoder uses the synchronization stream to find the current time value when either:
• The timestamp decoder exits reset.
• A forced synchronization event is broadcast by the timestamp encoder over the timestamp interface, for example because the timestamp value has changed.
When it has synchronized, the binary count channel provides information to enable subsequent timestamps to be returned accurately.
Because of this erratum, the timestamp decoder can synchronize to a timestamp value that is too small. This can be caused by a condition in the timestamp decoder itself, or by a condition in the narrow timestamp replicator, synchronous bridge or asynchronous bridge, which causes an error in the synchronization channel of the compressed timestamp stream.
Components connected to the timestamp decoder might receive incorrect subsequent timestamps, until after a period of time equal to the error in the timestamp value, after which the timestamp decoder returns the correct values. This is because the mechanism used by the binary count channel provides some resilience to errors in the synchronized time value.
When this erratum occurs, the error is often small and recovers in a small number of cycles. This is because the size of the error corresponds to the bit in the binary time value that was changing when the error occurred, and low-order bits change more frequently than high-order bits. However, the error might sometimes be large and the worst case error in the timestamp value is equal to the current timestamp value. After synchronizing to an incorrect value, the decoded timestamp output will count forward and may jump backwards before converging to the correct value.

Conditions

• Timestamps are communicated to a CoreSight component using a timestamp encoder and a timestamp decoder.
• Either the revision of the timestamp decoder is affected by this erratum, or the timestamp passes through a version of the narrow timestamp synchronous bridge or narrow timestamp asynchronous bridge that is affected by this erratum.

Implications

Timestamp values reported by CoreSight components might be incorrect after the component has been reset. Each timestamp decoder in the system can be affected independently.
If you have used the CoreSight timestamp decoder to provide timestamps for functional operation, for example the generic timer used by the processor, then this might have more significant consequences to the behavior of the system.

For example, the first timestamp reported after reset might be lower than the last timestamp reported before reset, and this can have a serious impact on operating system behaviour. Synchronization is typically not required in an active running system unless parts of the network are powered down.

Workaround

There is no generic workaround for this erratum. In some systems, it will be possible to compare the locally decoded timestamp with the main system timestamp to detect the error. If this can be done, performing a reset of the timestamp decoder will force it to re-synchronize.

337151

(806420) AXI-AP Debug Base Address Register returns upper and lower 32 bits swapped

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1 Fixed in r3p0.

Description
This erratum affects the following components:

- AXI Access Port.
    - cxdapaxiap
    - Component Revisions: r0p0, r0p1, r0p2

The AXI-AP Debug Base Address Register bits [63:32] and bits [31:0] are swapped. Bits [31:0] are returned in register offset 0xF0, and bits [63:32] are reported in register offset 0xF8, whereas the architecture for Memory Access Ports specifies that bits [31:0] should be reported in register offset 0xF8 and bits [63:32] should be reported in register offset 0xF0.
In 32-bit configurations of the AXI-AP, the debug base address is returned in register offset 0xF0, whereas it should be returned in register offset 0xF8. Register offset 0xF8 returns 0x00000000.

Conditions
A CoreSight ROM table is implemented that is accessed through the AXI-AP.

Implications
In 32-bit AXI-AP configurations, debug tools are likely to report that no ROM table is present even when a ROM table is present. This is because bit 0 of register 0xF8 indicates whether a ROM table is present, and this bit is zero.
In 64-bit AXI-AP configurations, debug tools might either report that no ROM table is present, or might report the incorrect address for the ROM table. If the wrong address is reported then they could access an illegal address when decoding the ROM table, potentially resulting in UNPREDICTABLE behavior.

Workaround
Debug tools should swap registers 0xF0 and 0xF8 in affected revisions of the AXI-AP. Later revisions of the AXI-AP should be accessed as normal.

334938

(814223) SWJDP sets parity error in Overrun mode

Status

Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r1p0, r2p0, r2p1, r3p0, r3p1, Fixed in r3p2

Description

This erratum affects the following components:

- SWJTAG Debug Port.
    - cxdapswjdp
    - Component Revisions: r1p2, r1p3, r1p4

The Serial Wire Debug Port (SW-DP) supports an overrun detection mode. When a transaction request is received when the previous transaction has not yet completed and a WAIT response is sent by the SW-DP, a sticky flag is set if overrun detection mode is enabled. The SW-DP then provides a FAULT response to subsequent transactions. In this mode, WAIT and FAULT responses require a data phase, but the data phase must be ignored.

As a result of this erratum, the SW-DP will perform a parity calculation and set the CTRL/STAT.WDATAERR bit at the end of the data phase for any transaction that generates a WAIT response. This occurs for both read and write accesses. When the data line is driven only by the pull-up, it is likely that the CTRL/STAT.WDATAERR bit will always be set when an overrun error occurs.

Conditions

1) The Serial wire debug protocol is in use.
2) Overrun detection is enabled (CTRL/STAT.ORUNDETECT ==1).
3) Two transaction requests are performed so that the second is received before the first has completed, causing a WAIT response.
4) The sticky bits in the CTRL/STAT register are read and WDATAERR is used as part of the function to determine which transactions must be replayed.

Implications

When the debug tools read the CTRL/STAT.WDATAERR bit as set, this should be interpreted as meaning that a previous write transaction failed to complete successfully. The debug tool must then identify and repeat the write transaction. Since this erratum results in CTRL/STAT.WDATAERR always being set after a WAIT response, it is likely that repeating the sequence of transactions from the earlier write access will again result in a similar WAIT response, preventing the debug tools from making progress in issuing a sequence of transactions.

Workaround

These workarounds apply to debug tools.
1) Avoid using Overrun Detection mode, or insert idle cycles between transaction requests to prevent WAIT responses from occurring.
2) The CTRL/STAT.WDATAERR bit can safely be ignored if the first transaction response observed was WAIT after an OK response. If the first response was ERR, it is not possible to determine if the CTRL/STAT.WDATAERR bit was set before the response was generated or in the overrun data phase.

323937

(811981) Narrow timestamp synchronization can be delayed

Status

Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0 Fixed in r3p1.

Description

This erratum affects the following components:

- Timestamp encoder.
  - cxtse
  - Component Revisions: r0p0, r0p1
- Narrow timestamp replicator.
  - cxntsreplicator
  - Component Revisions: r0p0
- Narrow timestamp decoder.
  - cxtsd
  - Component Revisions: r0p0, r0p1

The timestamp decoder component decodes the narrow timestamp stream into a full 64-bit timestamp. The narrow timestamp consists of the following channels:
• Synchronization channel: Periodically transmits the current timestamp value. Binary count channel: Gives timestamp information relative to previous timestamps.
The narrow timestamp decoder uses the synchronization channel to find the current time value when either:
• The narrow timestamp decoder exits reset.
• A forced synchronization event is broadcast by the timestamp encoder over the narrow timestamp interface, for example because the timestamp value has been reprogrammed.
When the decoder has synchronized, the binary count channel provides information to enable subsequent timestamps to be produced accurately. Because of this erratum, there are some scenarios when the timestamp decoder does not synchronize as rapidly as expected, and in some scenarios might never synchronize. When the decoder is not synchronized, it indicates that no timestamp is available to any component using the timestamp value. A timestamp value of zero from the timestamp decoder indicates no timestamp is available.
The timestamp encoder continuously generates synchronization sequences, and the decoder continues to use this information until it becomes synchronized.

Conditions

• Timestamps are communicated to a component using a timestamp encoder and a timestamp decoder (connected directly, or using other timestamp components).
• Any of the following apply:
1) The synchronization sequence is the first synchronization generated by the encoder after it has been reprogrammed.
2) The count value presented to the encoder is static. This occurs when the timestamp generator has been re-programmed but counting is not enabled.
3) The count value does not increment by one on every cycle at the encoder.
4) A narrow timestamp replicator is used.

Implications

Because of this erratum, the synchronization of a timestamp decoder will take longer than expected for a synchronization sequence to propagate from encoder to decoder. For systems where the CoreSight Timestamp Generator connects directly to the timestamp encoder, and where the narrow timestamp replicator is not used, synchronization will only be delayed by up to 65 cycles of the slowest clock in the timestamp system.
For systems that use the narrow timestamp replicator, or that present a count value to the timestamp encoder which does not increment by exactly one every cycle, it is possible that timestamp synchronization will never occur.
When this erratum occurs, a timestamp value of zero is output by the narrow timestamp decoder. For CoreSight timestamps, this indicates to debug tools that the timestamp value is not known.

Workaround

If a system affected by this erratum is used for generic time, the operating system boot code must be aware that the time value might be zero when it is first read. If the read value is zero, the read must be repeated until the value is not zero. A side effect of this workaround is that the boot time might be delayed. This workaround is not effective in systems where synchronization never occurs.

For CoreSight timestamping, there is no workaround for this erratum.

209972

(771719) CTIINTACK register needs clearing each time it is set

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category B
Fault Status: Present in: r0p0 Fixed in r1p0.

Description
This erratum affects the following components:

- Cross Trigger Interface.
  - CSCTI
  - Component Revisions: r0p3

The CTI contains a CTIINTACK register, which enables a trigger to be acknowledged through software, instead of using a hardware knowledge using the CTITRIGOUTACK input. The correct operation of this register is that writing a one to the bit corresponding to a trigger output will cause that trigger to be cleared, and this will not affect future triggers.
Because of this erratum, when a bit in the CTIINTACK register is set, it remains set until cleared by writing zero to the register. This causes the corresponding trigger outputs to be acknowledged immediately if they occur again, which can lead to them being missed.
The CTIINTACK register is normally used in two cases:
• To clear a debug-originated interrupt, if required by the interrupt controller.
• To clear a debug entry request generated by another processor, when cross-halting is used.

Conditions
The following conditions must occur:
• A CTI trigger output fires.
• The CTI CTIINTACK register is used to acknowledge the trigger output, by writing a one to the bit corresponding to that trigger output.
• The same trigger output fires again before the corresponding bit in the CTIINTACK register is cleared.

Implications
Trigger outputs might be missed:
• In the case of a debug-originated interrupt that uses CTIINTACK to clear the interrupt, events other than the first event might not cause an interrupt to occur.
• In the case of a cross-halting debug request, after the first time a processor halts and restarts, it might halt without halting other processors with it.

Workaround
This is a workaround for tools vendors.
When the CTIINTACK register is written with a nonzero value, it must be immediately written to again with the value zero. This prevents any future events on the corresponding trigger output from being acknowledged.
If this workaround is used, there remains a race condition, whereby a trigger output occurring between the two register writes might be lost. This is in general not significant, because the timing of trigger outputs and the timing of register writes are not highly correlated, and if the trigger output had occurred before the first register write, then it would also have been lost.

Category B (rare)

There are no errata in this category.

Category C

1624041

AHB-AP can issue transactions where HADDR[1:0] is not aligned to HSIZE on the AHB

Status

Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer CatC
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0, r3p1, r3p2, r3p2-50, Fixed in Open.

Description
This erratum affects the following components:

- AHB Access Port.
  - cxdapahbap
  - Component Revisions: r0p5, r0p6, r0p7, r0p8, r0p9

The ARM Debug Interface v5 Architecture Specification specifies a TAR (Transfer Address Register) in the MEM-AP that holds the memory address to be accessed.

TAR[1:0] is used to drive HADDR[1:0] when accesses are made using the Data Read/Write register DRW.

When the AHB-AP is programmed to perform a word or half-word sized transaction the AHB-AP does not force HADDR[1:0] to be aligned to the access size. This can result in illegal AHB transactions that are not correctly aligned according to HSIZE if HADDR[1:0] is programmed with an unaligned value.

Conditions
1) TAR[1:0] programmed with a value that is not aligned with the size programmed in the CSW register of the AHB-AP.
2) An access is initiated by an access to the Data Read/Write Register (DRW) in the AHB-AP.

Implications
As a result of the programming conditions listed above, AHB-AP erroneously initiates an access on the AHB with HADDR[1:0] not aligned to the size on HSIZE. This might initiate an illegal AHB access.

Workaround
TAR[1:0] must be b00 for word accesses, TAR[0] must be b0 for half-word accesses.
In normal usage models software programs the TAR with an address value that is aligned to the transaction size being made, so no workaround is necessary.

399409

(829169) ATB async bridge can generate spurious trace output after exit from low power state

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0, r3p1, Fixed in r3p2

Description
This erratum affects the following components:

- ATB asynchronous bridge.
  - cxatbasyncbridge
  - Component Revisions: r0p0, r0p1

It is possible for spurious trace data to be generated from the bridge on exit from a low power mode.

Conditions
The following sequence must occur:
1) A request is made on the LPI interface to put the bridge into a low power mode.
2) There is incoming trace data after the low power request. This is not usual as it is expected that upstream trace sources will be stopped before disabling the bridge).
3) There are insufficient enabled clock edges to complete the low power sequence before exiting the low power mode. This requires at least 1 enabled slave side clock edges (or a slave side reset) followed by at least 3 enabled master side clock edges (or a master side reset).
4) At least one of the following is true:

- The clocks on each side of the bridge are running asyncronously such that signals crossing the boundary may incur an extra cycle of latency compared to the synchroniser depth
- The synthesis tool has implemented the bridge such that delay(wrptr_grey) > delay(slave_safe_state)

Implications
Spurious trace data will be generated when the bridge leaves the low power mode.
If the slave side is reset then the data will have ATID set to zero, which means that trace might be captured on ID 0 in a trace sink. If the slave side is not reset the spurious data will have ATID set to the ID values of previous data transfers. In this case, it is expected that a debugger will need to search for a synchronization sequence in the trace streams and this erratum should have no effect.

Workaround
These are workarounds for system designers:

- Ensure that upstream trace is stopped before entering low power mode or
- Ensure that there is a least 1 enabled slave clock followed by 3 enabled master clocks before exiting low power mode.

341182

(813569) CSTPIU fails to output sync after the pattern generator is disabled in Normal mode

Status

Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1, r3p0, r3p1, r3p2, Fixed in Open

Description

This erratum affects the following components:

- Trace Port Interface Unit.
  - CSTPIU and cxtpiu
  - Component Revisions: r0p4, r0p5, r1p0

The TPIU includes a pattern generator that can be used to determine the operating behavior of the trace port and timing characteristics. This pattern generator includes a mode that transmits the test pattern for a specified number of cycles, and then reverts to transmitting normal trace data.
As a result of this erratum, when the TPIU is configured to operate in Normal Mode (FFCR.EnFCont==0), the synchronization sequence that is required between the test pattern and the trace data is not generated. Synchronization will be generated at later times as determined by the synchronization counter.

Conditions

The following conditions must all occur:

- The TPIU is configured in normal mode, FFCR.EnFCont==0
- The TPIU is configured with the formatter enabled, FFCR.EnFTC==1
- The pattern generator is enabled in timed mode, Current_test_pattern_mode.PTIMEEN==1

Implications

The timed mode of the TPIU is intended to permit the TPIU to transition between an initial synchronization sequence using the pattern generator and functional mode without any further programming intervention. If the synchronization sequence is not generated at the end of the test pattern, the trace port analyzer is unlikely to be able to capture the start of the trace stream correctly. Synchronization will be correctly inserted based on the value configured in the FSCR, once the specified number of frames of trace data have been output.

Workaround

This workaround requires software interaction to detect the completion of the test pattern sequence. In addition, any trace data present at the input to the TPIU is lost whilst the pattern generator is active. Any trace data present in the input to the TPIU before the formatter is re-enabled (and synchronization generated) will not be decompressible.
1) After enabling the pattern generator, set FFCR.StopOnFl==1 and FFCR.FOnMan==1.
2) Poll FFSR.FtStopped until 1 is read
3) Set FFCR.EnFTC==1

337157

(798174) Narrow timestamp synchronous bridge might reduce timestamp resolution

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0, r1p0 Fixed in r2p0.

Description
This erratum affects the following components:

- Narrow Timestamp Synchronous bridge.
  - cxntssyncbridge
  - Component Revisions: r0p0

In a timestamp distribution system the narrow synchronous bridge can only be used within a clock domain to bridge a master device with a slave device in the same clock domain.
The narrow timestamp synchronous bridge reduces resolution from Master to Slave device when the clken inputs on both slave and master interfaces are asserted high continuously.
All of the following conditions must occur
• The clkens input is always b1
• The clkenm input is always b1

Implications
The resolution of the distributed timestamp is reduced by half.
When clkens and clkenm inputs are not always high but are identical, the resolution is unchanged.
There is no impact to the functionality and the reconstructed timestamp always moves forward in time.

Workaround
There is no workaround for this erratum

328341

(771723) CoreSight SW-DP ignores data parity errors in writes to the ABORT register

Status

Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0 Fixed in r1p0.

Description

This erratum affects the following components:

- SWJTAG Debug Port.
  - cxdapswjdp
  - Component Revisions: r1p1

The ARM Debug Interface v5 Architecture Specification specifies the SWD protocol applies a simple parity check on packet requests or data transfers (read or write). If this parity check fails, WDATAERR bit becomes set in the CTRL/STAT register (RO, bit[7]).
The ARM Debug Interface v5 Architecture Specification also describes writing 1'b1 to either of the Debug Port ABORT register bits ORUNERRCLR, WDERRCLR, STKERRCLR, or STKCMPCLR will respectively clear either of the Debug Port CTRL/STAT register bits STICKYORUN, WDATAERR, STICKYERR and STICKYCMP.
When writing to the ABORT register in order to clear any of the CTRL/STAT sticky status bits, if a parity error is detected during the write, the sticky status bits value should remain unchanged except for WDATAERR that should be 1'b1.
If this erratum occurs, the selected sticky status bits may be cleared, thus ignoring the parity error detected during the write transfer.

Conditions

The following events must occur in the sequence defined:

1) At least one bit STICKYERR, STICKYCMP, or STICKYORUN in CTRL/STAT register are set from any of the following conditions:

- a transaction from the SWJ-DP to an AP transaction returns an error (STICKYERR becomes set) or,
- CTRL/STAT TRNMODE is set to 2'b10 (push verify) and the values compared by the DP do not match (STICKYCMP becomes set) or,
- CTRL/STAT TRNMODE is set to 2'b01 (push compare) and the values compared by the DP do match (STICKYCMP becomes set) or,
- CTRL/STAT ORUNDETECT is set to 1'b1 and the host debugger is issuing a request to the SWJ-DP followed by an acknowledge response that is not 'OK' (STICKYORUN becomes set).

2) The host debugger is writing 1'b1 to anyone bit ORUNERRCLR, STKERRCLR, STKCMPCLR of the ABORT register in an attempt to clear anyone of the matching bit STICKYERR, STICKYCMP, or STICKYORUN set in CTRL/STAT register.
3) During the ABORT register write (data transfer phase), a parity error is detected.

Implications

Error status bits may not be cleared as expected after a write to the ABORT register. Other debug-host state in the SWJ-DP is not affected. Use of the ABORT register and the occurrence of parity errors on the in interface are both rare.

Workaround

The debugger might be required to make multiple attempts to clear the sticky bit and needs to ensure that it has been cleared.

275600

(801369) APB-AP supports the ROM table only at 0x80000000

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0, r1p0, r2p0, r2p1 Fixed in r3p0.

Description
This erratum affects the following components:

- APB Access Port.
    - cxdapapbap
    - Component Revisions: r0p3, r0p4

The Debug Base Address Register (BASE) in the APB-AP (cxdapapbap) provides information on the presence and address of debug components connected to that AP. Typically, this register points to a ROM table, or it can point to a single debug component. Because of this erratum, this register contains a fixed value of 0x80000000 which indicates that the ROM table or debug component for this AP is present at 0x80000000. If the SoC implementation maps the ROM table or debug component to a different address, automatic discovery of the debug components will not work correctly.
Note: The debug components for a given AP can occupy a maximum of 2 GB of address space and are aliased to be accessible across the 4 GB address space. This means that the ROM table or the debug component mapped at address 0x80000000 is also visible at 0x00000000.
The tests provided with CoreSight SoC do not catch this erratum.
If the APB-AP is connected directly to the APBIC component provided with CoreSight SoC, then this erratum cannot occur.

Conditions
The erratum occurs in a system with all of the following attributes:
• The ROM table or debug component for the APB-AP is not located at address 0x80000000
• The system is not using the APBIC provided by CoreSight SoC.

Implications
When the conditions are met, if a debugger uses the BASE register to perform auto-discovery of debug components within the system, this will fail unexpectedly because the BASE register does not point to a valid ROM table or debug component.

Workaround
A workaround for system designers is to change the value of the BASE register. To do this, open the file:
1) coresight_soc/shared/logical/cxdapapbap/verilog/DAPApbApDefs.v
2) Edit the value of APB_ROM_ADDR in this file to the address of the top level ROM table or debug component in the system. This address must have:
• bit [31] set to 1b1
• bits [1:0] set to 2b11
For systems that cannot be modified, a workaround for tools vendors is to provide the debugger with the address of the ROM table or debug component for this AP, meaning the debugger does not use the APB-AP BASE register.

260244

(794419) Timestamp Encoder stops synchronizing the timestamp when the input counter stops counting

Status
Affects: CoreSight SoC - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0, r1p0 Fixed in r2p0.

Description
This erratum affects the following components:

- Timestamp Encoder.
    - cxtse
    - Component Revisions: r0p0

In a Timestamp distribution system the Encoder stops sending synchronization packets on the TSSYNC channel when the input count value stops counting.
If the Timestamp Generator stops counting due to either being disabled or halted then any slave devices which are not synchronized to the timestamp value must wait for the generator to resume its operation so that subsequent synchronization packets are generated. This introduces a delay before the slave device can synchronize to the latest timestamp value.

Conditions
The following condition must occur
1) The timestamp encoder input count value stops incrementing, which might happen because any of the following occur:
• The timestamp generator is disabled
• The timestamp generator is halted
The TSSYNC channel stops sending synchronization packets and only restarts sending synchronization packets when the input count value starts incrementing.

Implications
There is no impact to the reconstruction of the timestamp.
Timestamp receivers which are not synchronized to the current timestamp value will not begin the synchronization procedure until the timestamp generator starts counting. This might delay the receipt of the timestamp and prevent use of the timestamp until the synchronization procedure is complete.

Workaround
System Timestamp generators needs to be enabled for any slave device to sync-up to the new timestamp value.

209589

(771724) CTI Authentication Status register is incorrect

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0 Fixed in r1p0.

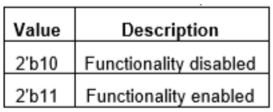
Description
This erratum affects the following components:

- Cross Trigger Interface.
  - CSCTI
  - Component Revisions: r0p3

The AUTHSTATUS register is a read-only register in the CTI that reports the debug level supported by the CTI and the current status of the debug level. The CoreSight Architecture Specification specifies bits [3:0] in the AUTHSTATUS register as below:

- [3:2] Non-Secure Non-Invasive Debug
- [1:0] Non-Secure Invasive Debug

For each of these fields, the value of the status bits as returned by the CTI and their meanings are:



Value	Description
2'b10	Functionality disabled
2'b11	Functionality enabled

In the CTI each pair of bits ([3:2] and [1:0]) in the AUTHSTATUS register currently read:

- When functionality is disabled - 2'b01

but should read (as per the Value and Description stated above):

- When functionality is disabled - 2'b10

The bits are swapped.

Condition 1
AUTHSTATUS[1:0] - Non-secure Invasive Debug

- DBGEN input to the CTI is LOW
- AUTHSTATUS register is read

Condition 2
AUTHSTATUS[3:2] - Non-secure non-Invasive Debug

- NIDEN input to the CTI is LOW
- DBGEN input to the CTI is LOW
- AUTHSTATUS register is read

Implications
The status of the debug level supported by the CTI as returned by the AUTHSTATUS register read is incorrect. The masking of trigger inputs and outputs using DBGEN and NIDEN is not affected by this erratum. The return of an incorrect value might lead to incorrect operation of debug tools.

Workaround
This is a workaround for users and tools vendors. When reading the AUTHSTATUS register, swap the bits in the affected fields and interpret the read data accordingly.

202988

(773919) The designer of AXI-AP is incorrectly identified by the identification register

Status
Affects: CoreSight SoC-400 - Perpetual
Fault Type: Programmer Category C
Fault Status: Present in: r0p0 Fixed in r1p0.

Description
This erratum affects the following components:

- AXI Access Port.
  - cxdapaxiap
  - Component Revisions: r0p0

The JEDEC code field which provides the information on the designer of the component in the identification register (IDR) of AXI-AP has a wrong value of 0x38 which is not the correct identification code for ARM. The correct value should be 0x3B.

Conditions
A read of the IDR register of AXI-AP.

Implications
This erratum may lead to wrong identification of designer of the component. However, the component can still be identified as AXI-AP and should have minimal impact on tools operation.
This has no functional effect on the operation of the AXI-AP.

Workaround
There is no workaround for this erratum.
When a debugger read to the IDR register returns the value 0x04710004, the debugger must consider that the designer field is incorrect and there is a probability of this being an ARM designed AXI-AP.