



CoreSight™ SoC-600M (TM250)

Software Developer Errata Notice

Date of issue: 08-Jul-2022

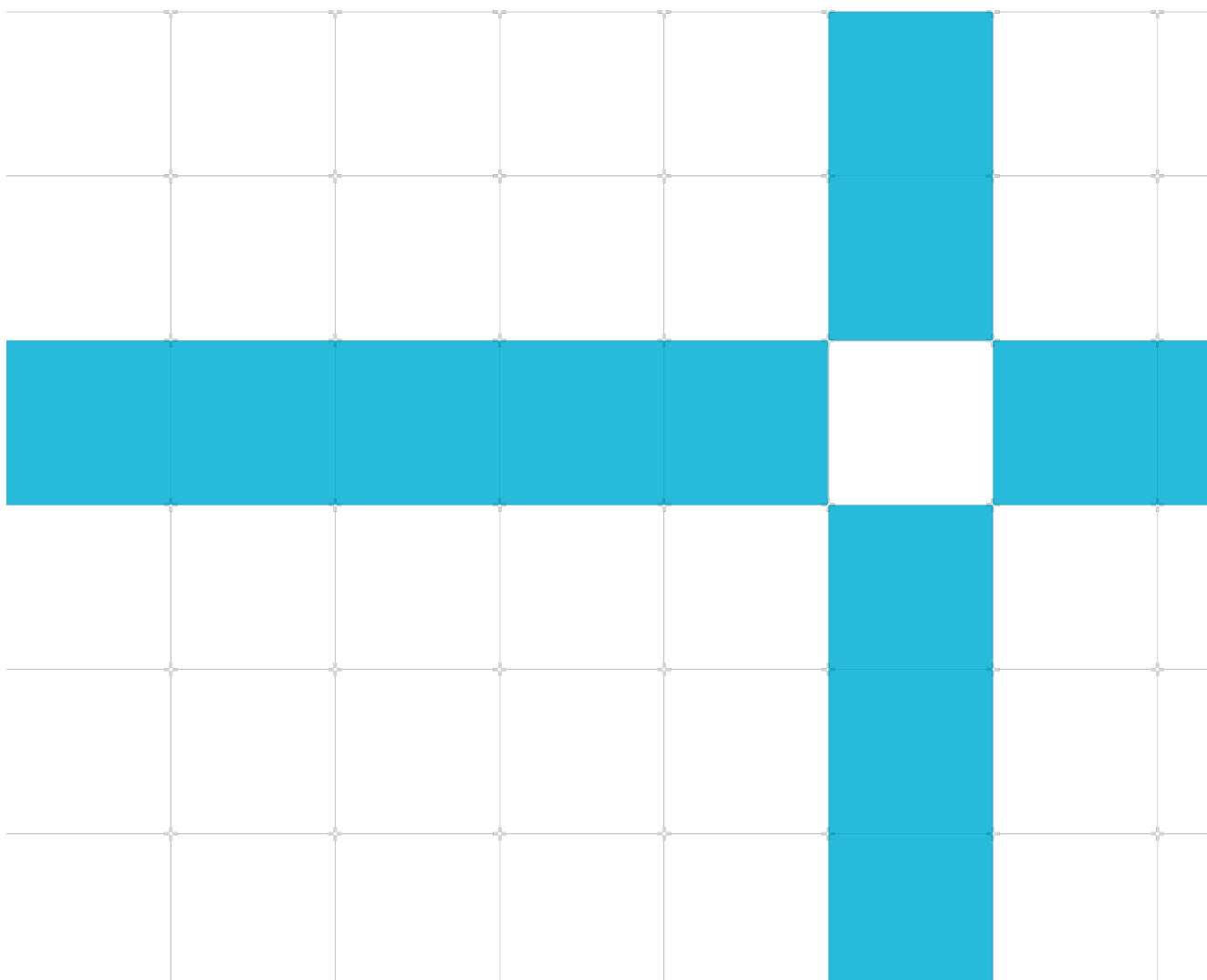
Non-Confidential

Document version: v5.0

Copyright © 2019-2022 Arm® Limited (or its affiliates). All rights reserved.

Document ID: SDEN-1527804

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 © 2019-2022 Arm® Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality status

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

Unrestricted Access is an Arm internal classification.

Product status

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on CoreSight™ SoC-600M (TM250), 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.

Contents

Introduction	5
Scope	5
Categorization of errata	5
Change Control	6
Errata summary table	7
Errata descriptions	8
Category A	8
Category A (rare)	8
Category B	9
1693612 TPIU generates spurious data when stopped	9
Category B (rare)	10
Category C	11
1594083 TPIU stops accepting trace when the trigger counter is written	11
1594082 TPIU stops accepting trace when FFCR is written	13

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](#) identifies errata that have been fixed in each product revision.

08-Jul-2022: Changes in document version v5.0

No new or updated errata in this document version.

28-Jun-2021: Changes in document version v4.0

No new or updated errata in this document version.

07-Oct-2020: Changes in document version v3.0

No new or updated errata in this document version.

19-Jun-2020: Changes in document version v2.0

ID	Status	Area	Category	Summary
1693612	Updated	Programmer	Category B	TPIU generates spurious data when stopped
1594082	Updated	Programmer	Category C	TPIU stops accepting trace when FFCR is written
1594083	Updated	Programmer	Category C	TPIU stops accepting trace when the trigger counter is written

14-Dec-2019: Changes in document version v1.0

ID	Status	Area	Category	Summary
1693612	New	Programmer	Category B	TPIU generates spurious data when stopped
1594082	New	Programmer	Category C	TPIU stops accepting trace when FFCR is written
1594083	New	Programmer	Category C	TPIU stops accepting trace when the trigger counter is written

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
1693612	Programmer	Category B	TPIU generates spurious data when stopped	r0p0	r1p0
1594083	Programmer	Category C	TPIU stops accepting trace when the trigger counter is written	r0p0	r1p0
1594082	Programmer	Category C	TPIU stops accepting trace when FFCR is written	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

1693612

TPIU generates spurious data when stopped

Status

Affects: CoreSight SoC-600M
Fault Type: Programmer Category B
Fault Status: Present in: r0p0, Fixed in: r1p0

Description

This erratum affects the following components:

- Trace Port Interface Unit (TPIU)
 - css600_tpiu
 - Component Revisions: r0p0, r1p0

The TPIU generates a **tracectl** output for compatibility with legacy Trace Port Analyzers (TPAs). The **tracectl** pin is not required when the TPIU operates in continuous mode, which should be supported by all modern TPAs. Arm recommends that the **tracectl** pin is not exported off chip, thus saving an IO pin.

As a result of this erratum, the TPIU outputs h0001 on **tracedata[]** when stopped. If the TPA does not see the **tracectl** pin it interprets the static **tracedata[]** value as further trace packets.

Conditions

1. The TPIU trace port interface to the TPA does not include the **tracectl** pin and is programmed to operate in Continuous mode.
2. The TPIU is programmed to stop, and enters the Stopped state.

Implications

When the TPIU stops the TPA will receive the expected trace data followed by a continuous stream of spurious trace packets.

When CSPSR is configured for 1 bit, **tracedata[0]** is static 1, which is interpreted as ATID=h7F, a reserved value.

When CSPSR is configured for N=2..32 bits, **tracedata[N-1:0]** is h0001.

This can be interpreted as new trace packets, starting with a static or changing trace ID, followed by static or changing trace data, followed by a static or changing auxiliary byte.

- When CSPSR is configured for 1 bit: Spurious trace data with ATID=h7F (reserved value).
- When CSPSR is configured for 2 bits: Spurious trace data with ATID=h00 and ATID=h2A

- When CSPSR is configured for 3 bits: Spurious trace data with ATID=h00 and ATID=h24
- When CSPSR is configured for 4 bits: Spurious trace data with ATID=h00 and ATID=h08
- When CSPSR is configured for 5 bits: Spurious trace data with ATID=h00 and ATID=h10
- When CSPSR is configured for 6 bits: Spurious trace data with ATID=h00 and ATID=h20
- When CSPSR is configured for 7 bits: Spurious trace data with ATID=h00 and ATID=h40
- When CSPSR is configured for 8 to 32 bits: Spurious trace data with ATID=h00

Workaround

When a trigger would be used to cause the TPIU to stop, debug tools need to program the TPIU:

- To not stop, by ensuring FFCR.StopTrig=0 and FFCR.StopFl=0.
- To generate a flush on a trigger event (FFCR.FOnTrig=1),
- To generate a trigger on flush completion (FFCR.TrigFl=1).

The debug tools need to detect the trigger generated on the trace port and then decide to stop trace capture. The debug tool might need to wait for an amount of time or data to ensure enough data has been captured from the TPIU.

If a trigger packet was inserted on ATB, the debug tool has to deal with two triggers on the trace port. In such a scenario, on detecting the first trigger the debug tool should wait for an amount of time or data to ensure enough data has been captured from the TPIU, although this might not guarantee all flushed data is captured.

When any other condition would be used to stop the TPIU, the debug tool might need to manually stop capture of trace.

Category B (rare)

There are no errata in this category.

Category C

1594083

TPIU stops accepting trace when the trigger counter is written

Status

Affects: CoreSight SoC-600M

Fault Type: Programmer Category C

Fault Status: Present in: r0p0, Fixed in: r1p0

Description

This erratum affects the following components:

- Trace Port Interface Unit.
 - css600_tpiu
 - Component Revisions: r0p0, r1p0

With this erratum the TPIU stalls the ATB slave interface indefinitely.

Conditions

The following conditions must exist:

- The TPIU register FFCR.EnFCont is 1, which enables Continuous Formatting Mode.
- The TPIU register TCVR was written with the value 1 or higher, which enabled trigger byte counting.
- The TPIU register bit STMR.TrgRun is 1. The trigger byte counter was loaded with the TCVR value and the trigger byte counter decrements depending on the trace traffic on the ATB slave interface.
- The TPIU register bit FFCR.StopTrig is 0.
- The TPIU register bit FFCR.StopFI is 1 or 0.
- The generation of upstream flush requests is enabled by having set at least one of the TPIU register bits FFCR.FonMan, FFCR.FonTrig or FFCR.FonFIIn, and an upstream flush is initiated by one of these mechanisms.
- The upstream flush is subsequently acknowledged by the system and the TPIU proceeds to drain internal buffers.
- The TPIU register TCVR is written with the value zero before the TPIU has drained its internal buffers.

Implications

The following implications persist until the TPIU is reset:

- The ATB slave interface is stalled, accepting no more trace.

- The TPIU does not stop, and FFSR.FtStopped is never set to 1.
- FFSR.FIInProg remains at the value 1.

Workaround

Debug tools must not write to TCVR while the TPIU is not stopped, and must only change TCVR when FFSR.FtStopped==1.

1594082

TPIU stops accepting trace when FFCR is written

Status

Affects: CoreSight SoC-600M
Fault Type: Programmer Category C
Fault Status: Present in: r0p0, Fixed in: r1p0

Description

This erratum affects the following components:

- Trace Port Interface Unit.
 - css600_tpiu
 - Component Revisions: r0p0, r1p0

With this erratum the TPIU stalls the ATB slave interface indefinitely.

Conditions

The following conditions must exist:

- TPIU register FFCR.EnFCont is 1, which enables Continuous Formatting Mode.
- At least one of the TPIU register bits FFCR.StopTrig or FFCR.StopFl is 1.
- A condition occurs which causes the TPIU to stop capture.
- Before the TPIU stops capture, the debugger clears TPIU register bits FFCR.StopTrig and FFCR.StopFl to 0.

Implications

The following implications persist until the TPIU is reset:

- The ATB slave interface is stalled, accepting no more trace.
- The TPIU does not stop, and FFSR.FtStopped is never set to 1.
- FFSR.FInProg remains at the value 1.

Workaround

Debug tools must avoid clearing FFCR.StopTrig and FFCR.StopFl while the TPIU is not stopped, and must only change these fields when FFSR.FtStopped==1.

Alternatively, disable the trigger indication on the trace port by clearing FFCR.TrigEvt and FFCR.TrigIn before clearing FFCR.StopTrig and FFCR.StopFl.

