Arm® Corstone™ SSE-710 Subsystem

Software Developer Errata Notice

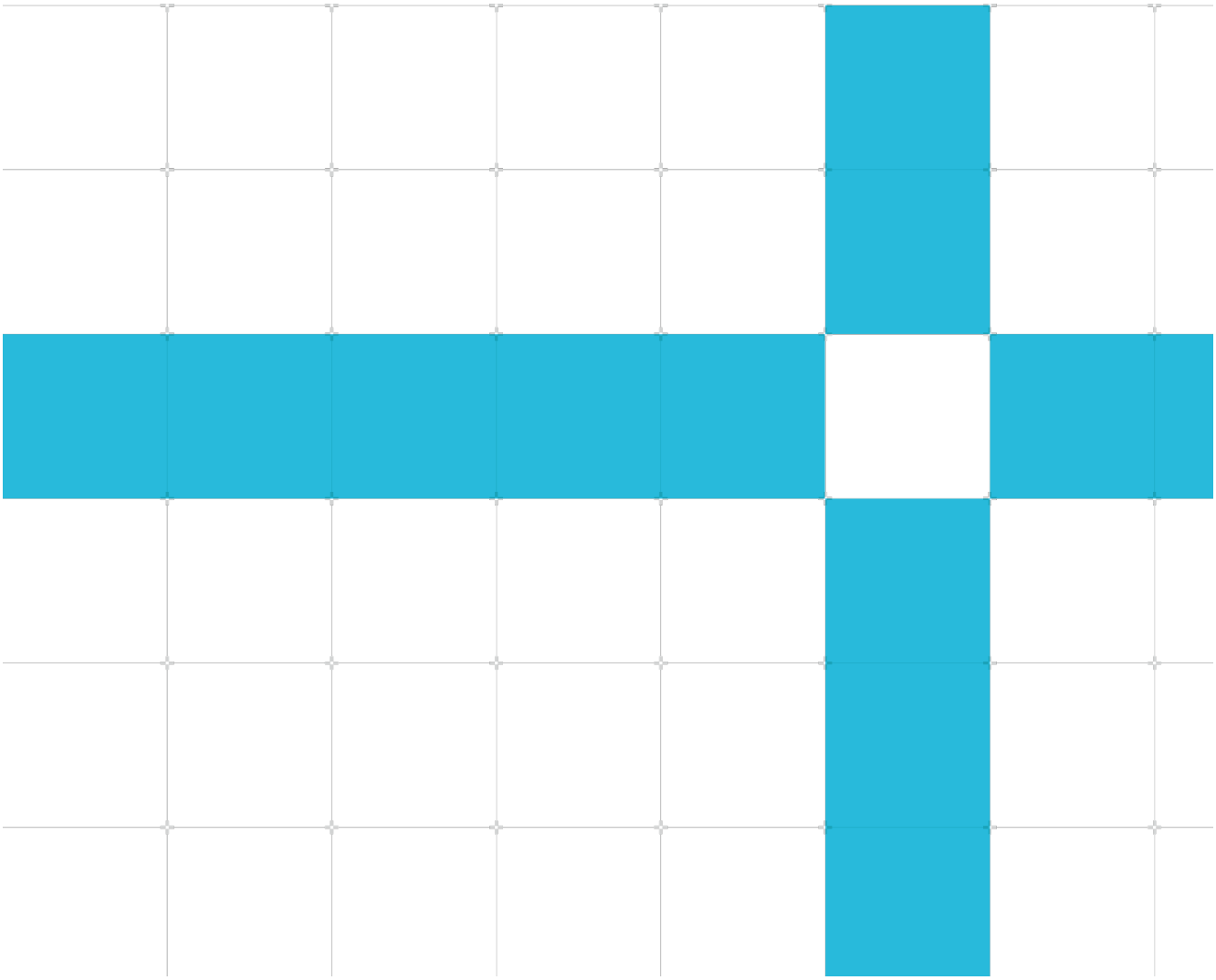
Date of issue: 01-Jul-2022

Non-Confidential

Document version: v2.0

Document ID: SDEN-2142076

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.

Unrestricted Access is an Arm internal classification.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® Corstone™ SSE-710 Subsystem, 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

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.

01-Jul-2022: Changes in document version v2.0

ID	Status	Area	Category	Summary
2142118	Updated	Programmer	Category B	System Generic Timer - CNTFRQ register in CNTBaseN and CNTCTLBase views inconsistent

09-Apr-2021: Changes in document version v1.0

ID	Status	Area	Category	Summary
2142118	New	Programmer	Category B	System Generic Timer - CNTFRQ register in CNTBaseN and CNTCTLBase views inconsistent

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
2142118	Programmer	Category B	System Generic Timer - CNTFRQ register in CNTBaseN and CNTCTLBase views inconsistent	r0p0	r0p0

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

2142118
System Generic Timer - CNTFRQ register in CNTBaseN and CNTCTLBase views inconsistent

Status

Affects: CG071 - Corstone-710 Subsystem

Fault Type: Programmer Cat B

Fault Status: Present in r0p0 EAC. Fixed in r0p0 REL

Description

1. The behavior of Generic Timer register CNTFRQ has been changed for Armv8-A.
2. In the current Armv7-A implementation, the register is visible in two frames CNTBaseN and CNTCTLBase, which are implemented as independent registers. In Armv8-A these registers are linked and reflect the same value.
3. Software that expects the Armv8-A behavior: writes the expected value to the CNTFRQ register in the CNTCTLBase frame and then expects this value to be reflected when the value is read from the CNTFRQ register in the CNTBaseN frame.

However, as these registers are independent in the Armv7-A implementation, the values are not reflected.

Configurations affected

All configurations are affected.

Conditions

1. Software writes a value to the CNTFRQ register through the CNTCTLBase frame
2. Software reads the CNTFRQ register through the CNTBaseN frame
3. The value of CNTFRQ read via the CNTBaseN frame does not reflect the value written via the CNTCTLBase frame as these are implemented as independent registers.**

Implications

OS software might fail to boot due to inconsistencies in the CNTFRQ views.

Workaround

In the current Armv7-A implementation, although the CNTFRQ is normally 'Read-Only', for initial configuration it can be written through the CNTBaseN frame.

Therefore, software must write the required CNTFRQ value to both the CNTBaseN and CNTCTLBase frames.

This ensures consistency when reading the CNTFRQ value from either CNTBaseN or CNTCTLBase frames.

For example:

mmio_write_32(ARM_SYS_TIMCTL_BASE + CNTCTLBASE_CNTFRQ, freq_val);

mmio_write_32(ARM_SYS_CNT_BASE_NS + CNTBASEN_CNTFRQ, freq_val);

Category B (rare)

There are no errata in this category.

Category C

There are no errata in this category.