Arm® CoreLink™ NI-710AE Network-on-Chip Interconnect

Software Developer Errata Notice

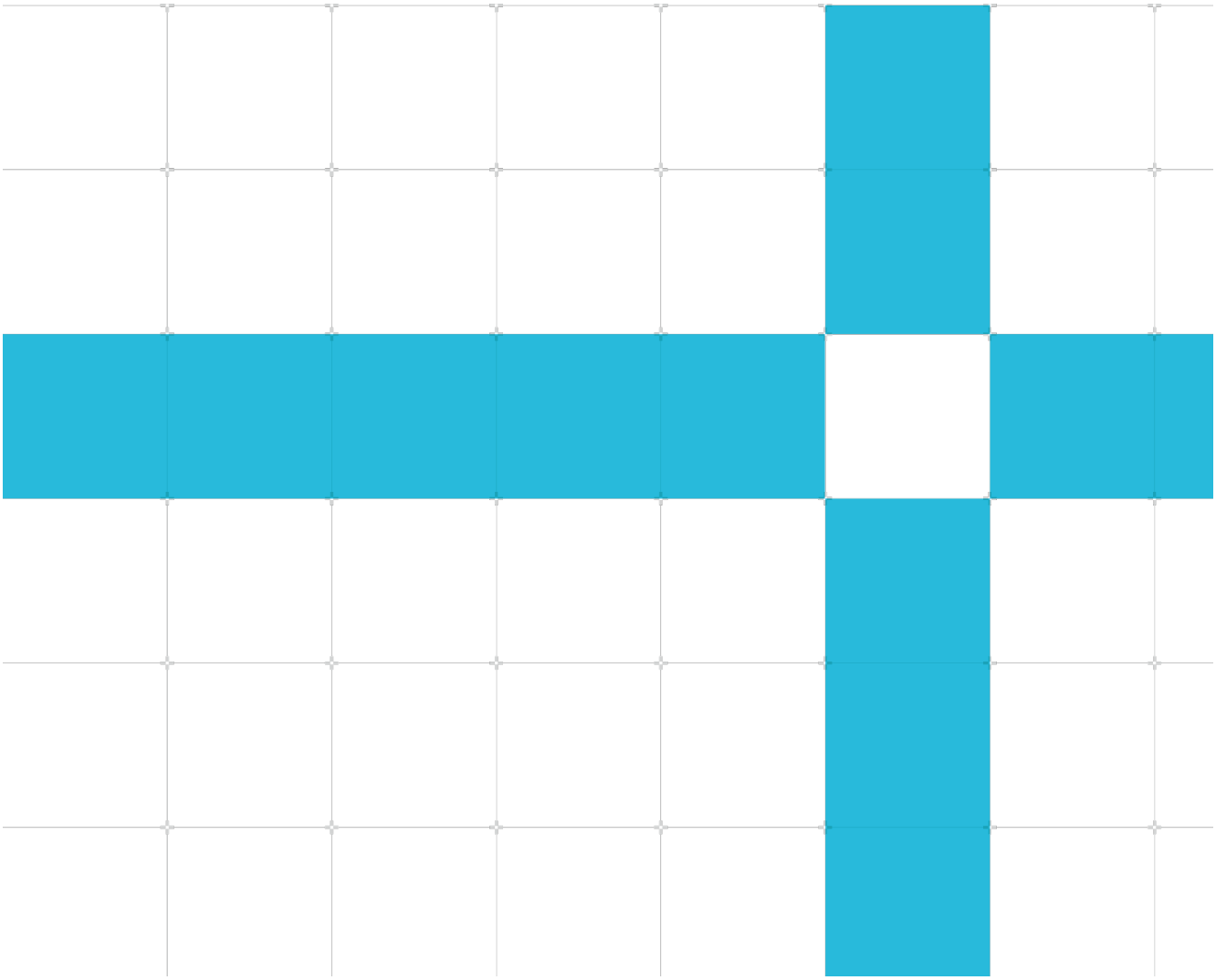
Date of issue: June 27, 2024

Non-Confidential

Document version: 5.0

Document ID: SDEN-2859927

This document contains all known errata since the r0p1 release of the product.



This document (SDEN_2859927_5.0_en) was issued on June 27, 2024.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-2859927**

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**.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® CoreLink™ NI-710AE Network-on-Chip 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**.

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.

June 27, 2024: Changes in document version v5.0

No new or updated errata in this document version.

March 08, 2024: Changes in document version v4.0

ID	Status	Area	Category	Summary
3193266	New	Programmer	Category C	Region count field in peripheral_id4 configuration register is incorrect.

December 04, 2023: Changes in document version v3.0

No new or updated errata in this document version.

June 30, 2023: Changes in document version v2.0

ID	Status	Area	Category	Summary
2982194	New	Programmer	Category C	Overflow reported for single destination ID checker error events on GT and AUB

February 24, 2023: 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
2982194	Programmer	Category C	Overflow reported for single destination ID checker error events on GT and AUB	r0p1	Open
3193266	Programmer	Category C	Region count field in peripheral_id4 configuration register is incorrect.	r0p1	Open

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

There are no errata in this category.

Category B (rare)

There are no errata in this category.

Category C

2982194
Overflow reported for single destination ID checker error events on GT and AUB

Status

Affects: PL627
Fault Type: CAT C - Programmer
Fault Status: Present in: r0p1. Fixed in: Open

Description

A single DESTID error in a NI-710AE configuration may be reported for multiple cycles, causing a false error overflow indication in the FMU (Fault Management Unit). This may occur on the GT and AUB networks (SM 15 and 16).

Configurations Affected

This issue happens in configurations where all the following conditions are true:

- NI-710AE has DESTID checking enabled though internal network protection.

Conditions

The following conditions are required to hit this issue:

- A GT/AUB packet is sent to an endpoint with a corrupted TGTID
- A GT/AUB packet is incorrectly routed to the wrong endpoint

Implications

When a DESTID error is encountered, the error may be reported for several cycles and cause an overflow to be falsely indicated in the FMU_ERR_STATUS.OF field.

Error handling software may use the OF indicator to count the number of errors to distinguish between transient faults and permanent faults. The action taken for a transient fault may be to reset the device, whereas the action taken for a permanent fault may be to disable the device.

Workaround

The error overflow field may be incorrect when a DESTID (SM 15 and 16) error is encountered. The overflow field is accurate as long as there is not a destination ID error present. For SM 15 and 16, FuSa software should ignore the OF bit, and instead use a software counter to determine the frequency and severity of events.

3193266
Region count field in peripheral_id4 configuration register is incorrect

Status

Affects: PL627
Fault Type: CAT C - Programmer
Fault Status: Present in: r0p1. Fixed in: Open.

Description

The peripheral_id4 software configuration register in the Global register block includes a read-only field named region_count that contains a configuration-dependent value. The calculation that sets the value of the region_count field is erroneous and so it should be assumed that the value of the field is incorrect.

The value of the peripheral_id4.region_count field is intended to specify, as a binary logarithm, the number of register blocks that are occupied for the interconnect programmers view. When a configuration is rendered, the rendering process calculates the region_count value and sets straps in the rendered RTL for the region_count field. The calculation depends on parameters such as the number of:

- Endpoints
- Subfeatures enabled for an endpoint
- Voltage domains
- Clock domains

However, the calculation counts the number of programming blocks incorrectly and underreports the number of pages in the configuration space. Although there are circumstances in which the field value is correct, in the majority of cases, the field value is calculated incorrectly. As a result, it should always be assumed that the value that is reported in the peripheral_id4.region_count field is incorrect.

Configurations affected

This issue occurs in all configurations.

Conditions

This issue occurs under all conditions and is always present.

Implications

Because the peripheral_id4.region_count field can report incorrect values, software should never use this field, either during the software discovery process or when determining the amount of address space required for the interconnect programming model in the system address map.

Workaround

The following independent workarounds can be used to avoid this issue:

- Use the software discovery process that is described in the product TRM to determine the components that are in the product configuration. This process enables software to discover the offset in the configuration address map for each component, how many pages the components use, and the total address space required. Discovery is a standard flow that can be used by software and is not just a workaround for this particular issue.
- Use the configuration address space value that is calculated by Socrates when the product configuration is rendered. If this value is required by system configuration software before discovery takes place, or if discovery is not used, the value should be stored in a software-accessible table. Such a table could be part of the information that is used by the device driver for the interconnect.

Proprietary notice

(PRE-1121-V1.0)

Product and document information

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

Product status

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

Product completeness status

The information in this document is Final, that is for a developed product.

Product revision status

The rxpy identifier indicates the revision status of the product described in this manual, where:

rx

Identifies the major revision of the product.

py

Identifies the minor revision or modification status of the product.