Arm® Cortex®-R8 MPCore

Product Revision r0

Software Developers Errata Notice

Non-Confidential - Released

Software Developers Errata Notice

Non-Confidential Proprietary notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ

Web Address

http://www.arm.com

Feedback on content

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

- the document title
- the document number, PJDOC-466751330-9452
- the page numbers to which your comments apply
- a concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Release Information

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 in Chapter 2. Fixed errata are not shown as updated unless the erratum text has changed. The summary table in section 2.2 identifies errata that have been fixed in each product revision.

15 Feb 2019: Changes in Document v4

Page	Status	ID	Cat	Rare	Summary of Erratum
17	New	1366535	CatC		The debugger might hang in Lock mode if it accesses CPU1 registers

11 Sep 2018: Changes in Document v3

Page	Status	ID	Cat	Rare	Summary of Erratum
16	New	1258961	CatC		CS signals of SCU rams are not cleared when SCU Invalidate All command is complete

15 Mar 2016: Changes in Document v2

Page	Status	ID	Cat	Rare	Summary of Erratum
8	New	854531	CatB		Broadcasting invalidate instruction Cache by MVA may stall a core
9	New	854976	CatB		ETM sharing does not work with three or more cores
13	New	854322	CatC		CP15 read access to ITCM RAM location might return incorrect data value
14	New	854527	CatC		Eviction request missing in Tag RAM can incorrectly detect an ECC error in Data RAM
15	New	854921	CatC		"Correctable ECC errors on any bus" event does not take into account errors from the FPP of other cores

04 Dec 2015: Changes in Document v1

Page	Status	ID	Cat	Rare	Summary of Erratum
9	New	853621	CatC		DBGPRSR Sticky Reset status bit is set to 1 by the CPU debug reset instead of by the CPU non-debug reset
11	New	853622	CatC		Integration register shows incorrect behaviour
12	New	853623	CatC		UNDEFINED instructions might not raise an UNDEF exception

Contents

Chapter 1.
Introduction

This chapter introduces the errata notice for the Cortex-R8 processors.

1.1. Scope of this document

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

- the current status of the defect
- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a 'work-around' where possible

This document describes errata that may impact anyone who is developing software that will run on implementations of this Arm product.

1.2. Categorization of errata

Errata recorded in this document are split into the following levels of severity:

Table 1 Categorization of errata

Errata Type	Definition
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.

Chapter 2.
Errata Descriptions

2.1. Product Revision Status

The r*n*p*n* identifier indicates the revision status of the product described in this book, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

2.2. Revisions Affected

Table 2 below lists the product revisions affected by each erratum. A cell marked with **X** indicates that the erratum affects the revision shown at the top of that column.

This document includes errata that affect revision r0 only.

Refer to the reference material supplied with your product to identify the revision of the IP.

Table 2 Revisions Affected

ID	Cat	Rare	Summary of Erratum	r0p0	r0p1	r0p2	r0p3
854976	CatB		ETM sharing does not work with three or more cores	X			
854531	CatB		Broadcasting invalidate instruction Cache by MVA may stall a core	X			
1366535	CatC		The debugger might hang in Lock mode if it accesses CPU1 registers	X	X	X	
1258961	CatC		To update CS signals of SCU rams are not cleared when SCU Invalidate All command is complete	X	X		
854921	CatC		"Correctable ECC errors on any bus" event does not take into account errors from the FPP of other cores	X			
854527	CatC		Eviction request missing in Tag RAM can incorrectly detect an ECC error in Data RAM	X			
854322	CatC		CP15 read access to ITCM RAM location might return incorrect data value	X			
853623	CatC		UNDEFINED instructions might not raise an UNDEF exception	X	X	X	X
853622	CatC		Integration register shows incorrect behaviour	X	X	X	X
853621	CatC		DBGPRSR Sticky Reset status bit is set to 1 by the CPU debug reset instead of by the CPU non-debug reset	X	X	X	X

2.3. Category A

There are no errata in this category

2.4. Category A (Rare)

There are no errata in this category

2.5. Category B

854531: Broadcasting invalidate instruction Cache by MVA may stall a core

Category B

Products Affected: Cortex-R8.

Present in: r0p0

Description

Whenever a core executes a broadcasting invalidate Instruction Cache by MVA operation to another core, for example when copying executable code from flash to memory, the other cores flush and restart their prefetch unit each time without progressing. Because of this erratum, the other cores might stop their execution until there is no remaining broadcasting operation occurring anymore.

Configurations affected

This erratum affects all configurations of the processor with two or more cores.

Conditions

The erratum requires the following conditions:

- Two or more cores are working in SMP mode (by setting ACTLR.SMP to 1), and have the Cache maintenance broadcast enabled (by setting ACTLR.FW to 1).
- A core runs several successive invalidate Instruction Cache by MVA.
- The other cores, flushing and restarting their prefetch unit, need to execute some code that is either non-cacheable or cacheable, but that is neither in the Instruction Cache nor in the ITCM at this moment.

In this case, under certain timing circumstances, the other cores are not able to fetch the code they need to execute, and this might stop their execution.

Implications

Because of this erratum, if the interrupt handler code is either non-cacheable or cacheable, but is neither in the Instruction Cache nor in the ITCM at this moment, the other cores are not able to fetch it and, therefore, execute it. This increases the interrupt latency time until there is no longer any remaining invalidate Instruction Cache by MVA operation.

Workaround

A workaround for this erratum is to put critical interrupt code in the ITCM to have a reasonable interrupt latency.

854976: ETM sharing does not work with three or more cores

Category B

Products Affected: Cortex-R8.

Present in: r0p0

Description

The ETM contains logic, known as resources, that enables you to control tracing by specifying the exact set of triggering and filtering conditions required for a particular application. Resources include address comparators and data value comparators, counters, and a sequencer. In this erratum, the ETM cannot trace core 2 and core 3, regardless of the resource chosen.

Configurations affected

This erratum affects the configurations of the processor using the ETM sharing with three or more cores.

Conditions

The ETM is enabled and configured to trace core 2 and core 3.

In this situation, the triggering comparators do not fire as expected.

Implications

In this erratum, the ETM cannot trace core 2 and core 3. However, the ETM can trace core 0 and core 1.

Workaround

There is no workaround for this erratum.

2.6. Category B (Rare)

There are no errata in this category

2.7. Category C

853621: DBGPRSR Sticky Reset status bit is set to 1 by the CPU debug reset instead of by the CPU non-debug reset

Category C

Products Affected: Cortex-R8.

Present in: r0p0, r0p1, r0p2, r0p3

Description

DBGPRSR.SR, bit [3], is the Sticky Reset status bit. The ARM architecture specifies that the processor sets this bit to 1 when the non-debug logic of the processor is in reset state.

Because of this erratum, the processor sets this bit to 1 when the debug logic of the processor is in reset state, instead of when the non-debug logic of the processor is in reset state.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

The erratum requires no conditions.

Implications

Because of this erratum:

- DBGPRSR.SR might not be set to 1 when it should, when the non-debug logic of the processor is in reset state.
- DBGPRSR.SR might be set to 1 when it should not, when the debug logic of the processor is in reset state.

In both cases, the DBGPRSR.SR bit value might be corrupted, which might prevent the debug logic from correctly detecting when the non-debug logic of the processor has been reset.

Workaround

There is no work-around.

853622: Integration register shows incorrect behaviour

Category C

Products Affected: Cortex-R8.

Present in: r0p0, r0p1, r0p2, r0p3

Description

The ETM macrocell supports an integration test mode which permits direct control of some top-level pins to provide simple testing of connections between components.

The TRCITATBIDR register is intended to provide control of the ATIDMIx and ATIDMDx top-level pins when integration mode is enabled. If this erratum occurs, the module-level clock gating within the ETM prevents any update on these pins until a subsequent APB access is performed.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

This erratum occurs if all of the following conditions exist:

- Integration mode is enabled (TRCITCTRL.IME = 1)
- Software writes to TRCITATBIDR

Implications

Integration tests might expect to observe the side effect of writing to TRCITATBIDR and then checking a value in the connected ATB peripheral. Because of this erratum, no change is observed. This erratum does not affect the sequence recommended for topology detection.

The use of TRCITATBIDR is limited to testing during the development of a SoC.

Workaround

A work-around is to perform a second access to the APB (read or write) to any ETM register after any write to the TRCITATBIDR.

853623: UNDEFINED instructions might not raise an UNDEF exception

Category C

Products Affected: Cortex-R8.

Present in: r0p0, r0p1, r0p2, r0p3

Description

Executing some UNDEFINED opcodes might not raise an UNDEF exception; the processor will either execute an SDIV instruction or a NOP.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

The following opcodes are affected:

Thumb2: 1111 1011 1001 Rn (1)(1)(1)(1) Rd 0001 Rm

Implications

If the opcode is of the form:

Thumb2: 1111 1011 1001 Rn (1)(1)(1)(1) Rd 0001 Rm

then an SDIV instruction will be executed.

Otherwise, a NOP will be executed.

Note that this encoding is not part of the encoding that ARM guarantees will always UNDEF.

Workaround

The software should not rely on taking an UNDEF trap when executing an opcode within that encoding space. Note that this encoding is not part of the encoding that ARM guarantees will always UNDEF.

854322: CP15 read access to ITCM RAM location might return incorrect data value

Category C

Products Affected: Cortex-R8.

Present in: r0p0

Description

The Cache and TCM Debug Operation Register (CTDOR) allows to select either the Cache or the TCM. Then, subsequent accesses to the RAM Access Data Registers (RADRLO and RADRHI) allow to get to the returned value of the RAM. Because of this erratum, when the CTDOR selects the ITCM, the processor might not read the ITCM RAM location as expected, but might read an incorrect data value instead.

Configurations affected

This erratum affects all configurations of the processor that use ECC.

Conditions

The erratum requires the following conditions:

- The ECC is enabled on ITCM (because bit [10] of Auxiliary Control Register, ACTLR, is set to 1).
- The ITCMEER register contains a valid information.
- The CP15 read access hits the ITCMEER register.

Implications

Because of this erratum, it is not possible to analyze an erroneous RAM location in ITCM and determine whether an ECC error is a soft-error or a hard-error, as recommended in the basic ECC scheme described in the TRM.

Workaround

The recommended workaround for this erratum is to save the ITCMEER register and then clear it before analyzing an erroneous RAM location by using the following sequence:

MRC p15, 0, r0, c15, c5, 0; Read ITCM ECC entry

<save r0> (save ITCMEER contents)

BIC r0,r0,#0x1; Clear the valid bit of the ITCM ECC entry

MCR p15, 0, r0, c15, c5, 0; Write ITCM ECC entry

<Analyze the erroneous RAM location in ITCM>

<restore r0> (restore ITCMEER contents)

MCR p15, 0, r0, c15, c5, 0; Write ITCM ECC entry

854527: Eviction request missing in Tag RAM can incorrectly detect an ECC error in Data RAM

Category C

Products Affected: Cortex-R8.

Present in: r0p0

Description

When an ECC error occurs on the Data RAM of the Data Cache, the system is notified and then an automatic cache maintenance operation is performed.

This maintenance operation will write the Tag RAM of the Data Cache, to invalidate it, but will not write the Data RAM.

If a subsequent eviction is requested on the same index and way, it will read both Tag and Data RAMs, and report an ECC error from the Data part, even if it is missing in the Tag RAM. This will lead to the same error reported another time in the error bank, and on the external pin RAMERR.

Configurations affected

This erratum affects all configurations of the processor that use ECC.

Conditions

The erratum requires the following conditions:

- The ECC is enabled on caches (because bit [9] of Auxiliary Control Register, ACTLR, is set to 1).
- An ECC error is seen on the Data RAM of the Data Cache.
- One of the following conditions:
  - The error bank was already full on the first ECC error, OR
  - The error bank entry is cleared by software after this first ECC error.
- An eviction is requested on the same index and way as the previous ECC error, and detects a new ECC error. This eviction can be a natural eviction from a linefill or requested from a cp15 clean invalidate Data Cache by MVA operation.

Implications

Because of this erratum, it might be possible to report an ECC error twice to the system.

Workaround

As a workaround, when clearing the error bank entry, software can write to the Data RAM and the ECC Data RAM with valid values to remove any ECC error still present.

854921: "Correctable ECC errors on any bus" event does not take into account errors from the FPP of other cores

Category C

Products Affected: Cortex-R8.

Present in: r0p0

Description

The TRM describes the PMU event 0x67 "Correctable ECC errors on any bus".

These buses are the AXI master ports, the AXI slave ports, and the private FPP, one per core.

In this erratum, the ECC errors on the FPP of its dedicated core are counted, but not the ECC errors on the FPP of the other cores.

Configurations affected

This erratum affects all configurations of the processor with two or more cores using the FPP.

Conditions

The erratum requires the following condition:

- A correctable error occurs on the FPP of one core.

In this situation, the PMU counter of the other cores is not incremented.

Implications

The erratum alters the accuracy of the “Correctable ECC errors on any bus” event.

Workaround

There is no workaround.

1258961: CS signals of SCU rams are not cleared when SCU Invalidate All commands is complete

Category C

Products Affected: Cortex-R8.

Present in: r0p0, r0p1

Description

Writing to the SCU Invalidate All Register triggers the invalidation of all lines in the selected ways of the SCU duplicated tag RAMs. During this operation, the index of the selected RAMs is incremented, and the corresponding RAM CS signals are kept active-HIGH. When the last index is reached, CS signals are not cleared as expected and are stuck to HIGH, even if the invalidation sequence is complete.

Configurations affected

This erratum affects all r0p0 and r0p1 configurations of the processor. r0p2 is not affected as CS logic is different.

Conditions

1. The processor writes to the SCU Invalidate All Register (offset 0x0C from PERIPHBASE address). This starts an invalidation of all lines in the selected ways.
2. At the end of the invalidation, the CS signals corresponding to the selected ways are not cleared as expected. They are, however, cleared when a new access is performed to any duplicated tag RAMs.

Implications

It is not possible to rely on SCU CS signals to trigger a specific action, for example allowing a processor powerdown.

Workaround

No workaround is required for this erratum as this is not a functional issue.

However, if SCU CS signals need to be triggered, the workaround consists in forcing another SCU RAM access by executing for example a cacheable coherent LDR instruction after writing to the SCU Invalidate All Register.

1366535: The debugger might hang in Lock mode if it accesses CPU1 registers

Category C

Products Affected: Cortex-R8.

Present in: r0p0, r0p1, r0p2

Description

In Split/Lock configuration and when SAFEMODE pin is asserted HIGH (Lock mode), CPU1 is locked to CPU0 and they both behave as a single CPU.

However, as the ROM table always indicates the presence of 2 CPUs in the cluster, the debugger might attempt to access CPU1 registers through APB.

In this case, the access will never complete and the debugger will hang.

Configurations affected

This erratum affects the Split/Lock configuration, in Lock mode.

Conditions

1. The cluster is built with the Split/Lock configuration and runs in Lock mode.
2. The debugger performs an access to any CPU1 register accessible through the APB bus.

If the above conditions are met, then PREADYDBG is driven LOW which prevents the access to complete.

Implications

This erratum might cause the debugger to hang.

Workaround

There is no workaround to this erratum. The debugger must avoid accessing CPU1 registers in this case.

