

# ARM Processor Cortex<sup>TM</sup>-M0+

Product Revision r0

## Software Developers Errata Notice

Non-Confidential - Released

The ARM logo consists of the letters "ARM" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right of the letter "M".

---

## Software Developers Errata Notice

Copyright © 2012 ARM. All rights reserved.

### Proprietary notice

This document is protected by copyright and the practice or implementation of the information herein may be protected by one or more patents or pending 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.**

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.

Unless otherwise stated in the terms of the Agreement, this document is provided “as is”. ARM makes no representations or warranties, either express or implied, included but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement, that the content of this document is suitable for any particular purpose or that any practice or implementation of the contents of the document will not infringe any third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

Unless otherwise stated in the terms of the Agreement, to the extent not prohibited by law, in no event will ARM be liable for any damages, including without limitation any direct loss, lost revenue, lost profits or data, special, indirect, consequential, incidental or punitive damages, however caused and regardless of the theory of liability, arising out of or related to any furnishing, practicing, modifying or any use of this document, even if ARM has been advised of the possibility of such damages.

Words and logos marked with © or ™ are registered trademarks or trademarks, respectively, of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. Other brands and names mentioned herein may be the trademarks of their respective owners. Unless otherwise stated in the terms of the Agreement, you will not use or permit others to use any trademark of ARM Limited (or its subsidiaries).

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.

In this document, where the term ARM is used to refer to the company it means “ARM or any of its subsidiaries as appropriate”.

### Confidentiality status

This document is **Non-Confidential** but any disclosure by you is subject to you providing notice to and the acceptance by the recipient of, the conditions set out in this notice.

### 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](mailto:errata@arm.com). Give:

- the document title
- the document number, ARM-EPM-018879
- 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.

**20 Dec 2012: Changes in Document v4**

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

**18 Jul 2012: Changes in Document v3**

Page	Status	ID	Cat	Rare	Summary of Erratum
9	New	789069	CatC		A debugger write to the I/O port might be corrupted during a processor write

**03 May 2012: Changes in Document v2**

Page	Status	ID	Cat	Rare	Summary of Erratum
8	New	777019	CatC		Processor executing at HardFault priority can go into Lockup if an NMI occurs during a waited debug transaction

(First released version of this document was v2.0)

---

## Contents

<b>CHAPTER 1.</b>	<b>5</b>
<b>INTRODUCTION</b>	<b>5</b>
<b>1.1. Scope of this document</b>	<b>5</b>
<b>1.2. Categorization of errata</b>	<b>5</b>
<b>CHAPTER 2.</b>	<b>6</b>
<b>ERRATA DESCRIPTIONS</b>	<b>6</b>
<b>2.1. Product Revision Status</b>	<b>6</b>
<b>2.2. Revisions Affected</b>	<b>6</b>
<b>2.3. Category A</b>	<b>7</b>
<b>2.4. Category A (Rare)</b>	<b>7</b>
<b>2.5. Category B</b>	<b>7</b>
<b>2.6. Category B (Rare)</b>	<b>7</b>
<b>2.7. Category C</b>	<b>8</b>
777019: Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction .....	8
789069: A debugger write to the I/O port might be corrupted during a processor write .....	9

# Chapter 1.

## Introduction

This chapter introduces the errata notice for the ARM Cortex-M0+ processor.

### 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**

<b>Errata Type</b>	<b>Definition</b>
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 *mpn* identifier indicates the revision status of the product described in this book, where:

- rn* Identifies the major revision of the product.
- pn* 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	Revisions Affected	
				r0p0	r0p1
789069	CatC		A debugger write to the I/O port might be corrupted during a processor write	X	
777019	CatC		Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction	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**

There are no errata in this category

### **2.6. Category B (Rare)**

There are no errata in this category

## 2.7. Category C

### 777019: Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction

#### Category C

**Products Affected:** Cortex-M0+.

**Present in:** r0p0

#### Description

A debugger can perform memory accesses through the Cortex-M0+ processor bus matrix while the processor is running.

Because of this erratum, the processor might erroneously enter Lockup state if a debugger-initiated access on the AHB-Lite master port is subject to wait states while the processor is running, executing at HardFault priority and taking a Non Maskable Interrupt (NMI). Under very specific timing conditions, the processor might incorrectly stack a ReturnAddress of 0xFFFFFFFF on NMI entry. On NMI return, the processor unstacks the incorrectly stacked ReturnAddress and enters Lockup state at HardFault priority.

#### Configurations affected

The processor must be configured to both:

- Include debug.
- Perform most instruction fetches as word accesses.

In addition, the processor must be part of a system where both:

- The processor AHB-Lite master port can be subject to wait states.
- An NMI can be asserted to the processor.

#### Conditions

The following timing-specific conditions must all be met:

- The processor is running (not halted in Debug state) and is executing at HardFault priority.
- The processor executes a single-cycle instruction at a word-aligned address.
- The debugger performs an access through the AHB-Lite master port that is subject to wait states.
- An NMI becomes pending.

#### Implications

The processor stops executing the code in the HardFault handler and enters Lockup state at HardFault priority as if a fault had occurred.

#### Workaround

The debugger can work around this erratum by halting the processor in Debug state before performing accesses outside the Private Peripheral Bus (PPB) region of the memory map.



**789069: A debugger write to the I/O port might be corrupted during a processor write****Category C****Products Affected: Cortex-M0+.****Present in: r0p0****Description**

A debugger can perform memory accesses through the Cortex-M0+ processor bus matrix while the processor is running.

Because of this erratum, a debugger write to the I/O port might be corrupted if it occurs while the processor is executing a write. The processor write completes successfully. However, under specific timing conditions, the matrix might incorrectly replace the debugger write data with the value zero.

This erratum does not affect debugger writes outside the I/O port region of the memory map, or debugger reads.

**Configurations affected**

The processor must be configured to both:

- Include debug.
- Include the I/O port.

**Conditions**

The following timing-specific conditions must all be met:

- The processor is running (not halted in Debug state).
- The debugger performs a write within the I/O port region of the memory map.
- The processor performs a write.

**Implications**

The debugger might corrupt the targeted memory or configure the targeted device incorrectly.

**Workaround**

The debugger can work around this erratum by halting the processor in Debug state before performing writes to the I/O port region of the memory map.