ARM Core
Cortex™-M0 (AT510)
Errata Notice

This document contains all errata known at the date of issue in supported releases up to and including revision.r0p0 of Cortex-M0

Proprietary notice

Document confidentiality status

This document is Non Confidential.

Web address

http://www.arm.com/

Feedback on the product

If you have any comments or suggestions about this product, contact your supplier giving:

- The product name
- A concise explanation of your comments

Feedback on this document

If you have any comments on about this document, please send an email to support-cores@arm.com giving:

- The document title
- The documents number
- The page number(s) to which your comments refer
- A concise explanation of your comments

General suggestion for additions and improvements are also welcome.

Contents

Introduction

Scope

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

- a unique defect tracking identifier
- 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

Categorisation of Errata

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

Category 1	Behavior that is impossible to work around and that severely restricts the use of the product in all, or the majority of applications, rendering the device unusable.
Category 2	Behavior that contravenes the specified behavior and that might limit or severely impair the intended use of specified features, but does not render the product unusable in all or the majority of applications.
Category 3	Behavior that was not the originally intended behavior but should not cause any problems in applications.

Change Control

08 Oct 2010: Changes in Document v2

Page	Status	ID	Cat	Summary
10	New	735609	Cat 3	Watchpoint PC functions can report false execution
12	New	745383	Cat 3	Prefetch faulting instructions can erroneously trigger breakpoints

26 Mar 2009: Changes in Document v1

First revision of document with no errata listed

Errata Summary Table

The errata associated with this product affect product versions as below.

A cell shown thus **X** indicates that the defect affects the revision shown at the top of that column.

ID	Cat	Summary of Erratum	r0p0
735609	Cat 3	Watchpoint PC functions can report false execution	X
745383	Cat 3	Prefetch faulting instructions can erroneously trigger breakpoints	X

Errata - Category 1

There are no Errata in this Category

Errata - Category 2

There are no Errata in this Category

Errata - Category 3

735609: Watchpoint PC functions can report false execution

Status

Affects: product Cortex-M0.

Fault status: Cat 3, Present in: r0p0, Open.

Description

In the presence of interrupts including those generated by the SVC instruction, it is possible for both the data watchpoint unit's PC match facility and PC sample-register to operate as though the instruction immediately following the interrupted or SVC instruction had been executed.

Conditions

Either:

1. Halting debug is enabled via `C_DEBUGEN = 1`
2. Watchpoints are enabled via `DWTENA = 1`
3. A watchpoint is configured for PC sampling `DWT_FUNCTION = 0x4`
4. The same watchpoint is configured to match a "target instruction"
5. And either:
   - The "target instruction" is interrupted before execution, or
   - The "target instruction" is preceded by a taken SVC instruction
6. The DWT will unexpectedly match the "target instruction"
7. The processor will unexpectedly enter debug state once inside the exception handler

Or:

1. The debugger performs a read access to the `DWT_PCSR`
2. A "non-committed instruction" is preceded by a taken SVC instruction
3. The `DWT_PCSR` value unexpectedly matches the "non-committed instruction"

Implications

If halting debug is enabled and PC match watchpoints are being used, then spurious entry into halted debug state may occur under the listed conditions.

If the `DWT_PCSR` is being used for coarse grain profiling, then it is possible that the results can include hits for the address of an instruction immediately after an SVC instruction, even if said instruction is never executed.

Workaround

This errata does not impact normal execution of the processor.

A debug agent may choose to handle the infrequent false positive Debug state entry and erroneous PCSR values as spurious events.

745383: Prefetch faulting instructions can erroneously trigger breakpoints

Status

Affects: product Cortex-M0.

Fault status: Cat 3, Present in: r0p0, Open.

Description

External prefetch aborts on instruction fetches on which a BPU breakpoint has been configured, will cause entry to Debug state. This is prohibited by revision C of the ARMv6-M Architecture Reference Manual. Under this condition, the breakpoint should be ignored, and the processor should instead service the prefetch-abort by entering the HardFault handler.

Conditions

1. The Cortex-M0 implementation includes halting-mode debug support
2. The lowest 0.5GB of address space is capable of generating an AHB Error response
3. Halting debug is enabled via `CDEBUG_EN == '1'`
4. A BPU breakpoint is configured on an instruction in the first 0.5GB of memory
5. The fetch for said instruction aborts via an AHB Error response
6. The processor will erroneously enter Debug state rather than entering HardFault.

Implications

If halting debug is enabled and a BPU breakpoint is placed on an instruction with faults due to an external abort, then a non-compliant entry to Debug state will occur.

Workaround

This errata does not impact normal execution of the processor.

A debug agent may choose to avoid placing BPU breakpoints on addresses that generate AHB Error responses, or may simply handle the Debug state entry as a spurious debug event.