ARM Core
ARM966E-S r2 (AT200)
Errata Notice

This document contains all errata known at the date of issue in ARM966E-S r2 up to and including revision r2p1 of ARM966E-S. Refer to the Errata Summary Table on page 7 for information about which of the errata affect each revision.

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 email to mailto: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

12 Mar 2007: Changes in Document v12

Page	Status	ID	Cat	Summary
15	New	412551	Cat 3	Coprocessor 15 instructions immediately following an MSR instruction may be executed with the wrong privilege level

10 Feb 2004: Changes in Document v10

Page	Status	ID	Cat	Summary
14	New	318839	Cat 3	HLOCK asserted unnecessarily on AHB

22 Jan 2004: Changes in Document v9.5

Page	Status	ID	Cat	Summary
13	New	318838	Cat 3	xTCMCANCEL not asserted when all data is read from the TCM write buffer

14 Nov 2003: Changes in Document v9

Page	Status	ID	Cat	Summary
11	New	314712	Cat 2	ITCMSEQ incorrect during access to instruction TCM

23 Sep 2002: Changes in Document v7

Page	Status	ID	Cat	Summary
9	New	178312	Cat 2	Incorrect Data Tracing of STC Instruction

18 Jun 2002: Changes in Document v3

Page	Status	ID	Cat	Summary
10	New	227860	Cat 2	Incorrect clock input polarity on test wrapper lock-up latch

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	r2p0	r2p1
178312	Cat 2	Incorrect Data Tracing of STC Instruction	X	
227860	Cat 2	Incorrect clock input polarity on test wrapper lock-up latch	X	
314712	Cat 2	ITCMSEQ incorrect during access to instruction TCM	X	
326713	Cat 2	[(NTBSS) - ARM9E-S] Real monitor breakpoint missed in Thumb state	X	X
318838	Cat 3	xTCMCANCEL not asserted when all data is read from the TCM write buffer	X	
318839	Cat 3	HLOCK asserted unnecessarily on AHB	X	
412551	Cat 3	Coprocessor 15 instructions immediately following an MSR instruction may be executed with the wrong privilege level	X	X

Errata - Category 1

There are no Errata in this Category.

Errata - Category 2

178312: Incorrect Data Tracing of STC Instruction

Status

Affects: product ARM966E-S.

Fault status: Cat 2, Present in: r2p0, Fixed in r2p1.

Description

This erratum affects designs using an ETM and external coprocessors with the ARM966E-S.

The fault occurs when tracing STC instruction. When tracing this coprocessor instruction, the ARM966E-S sends incorrect data to the ETM, causing an incorrect trace output of the data bus. The data is stored correctly.

Conditions

The fault occurs when using the ETM to trace the data transferred by the coprocessor using an STC instruction. The instruction is correctly traced.

Implications

It is not possible to trace the data transferred by the STC instruction.

Workaround

This erratum can be worked around by inserting a load instruction to read back the data from the location to which it was stored.

227860: Incorrect clock input polarity on test wrapper lock-up latch

Status

Affects: product ARM966E-S.

Fault status: Cat 2, Present in: r2p0, Fixed in r2p1.

Description

There is a lock-up latch on the last output cell of the test wrapper scan chain. The purpose of the latch is to allow the WSO output to be held until the falling edge of the system clock. The WEDGE signal selects which signal appears on the WSO output; the wrapper output that changes on the positive edge of the system clock or the latch output that changes on the negative edge of the system clock.

The clock input on the test wrapper lock-up latch is the wrong polarity, so the WSO output changes on the positive edge when WEDGE is logic 1 or 0.

Conditions

The erratum affects implementations that require a test wrapper with the WSO output changing on the negative edge of the system clock.

Implications

The WEDGE signal cannot be used to select which signal appears on the WSO output.

Workaround

The WEDGE input must be tied to logic 1. No other action is required for a WSO output that changes on the positive edge of the system clock. A WSO output that changes on the negative edge of the system clock can be achieved by adding a latch external to the ARM966E-S macrocell.

314712: ITCMSEQ incorrect during access to instruction TCM

Status

Affects: product ARM966E-S.

Fault status: Cat 2, Present in: r2p0, Fixed in r2p1.

Description

While loading sequential sequences of some instruction types from the instruction TCM, the ITCMSEQ signal can be incorrectly de-asserted, falsely indicating that the address on ITCMADDR is not sequential to the previous access.

Conditions

The erroneous indication will occur in the following case:

- While running sequential Thumb code from instruction TCM, and when there was no memory access in the previous clock cycle.

Implications

The impact of this erratum is dependant on how the ITCMSEQ signal is used in the TCM system. If the ITCMSEQ signal is either used to determine wait-states (via ITCMWAIT), or to gate the memory chip-select (ITCMCS), then the performance of the design could be reduced.

Workaround

There is no workaround that can be implemented in the software running from ITCM.

326713: [(NTBSS) - ARM9E-S] Real monitor breakpoint missed in Thumb state

Status

Affects: product (NTBSS) - ARM9E-S, ARM966E-S.

Fault status: Cat 2, Present in: r2p0,r2p1, Open [Fixed in r2p1 of (NTBSS) - ARM9E-S].

Description

Under certain circumstances, a breakpoint on a Thumb instruction is not taken if Monitor mode debug is enabled.

In Monitor mode debug, if the core is configured for 32 bit Thumb instruction fetches, and a breakpoint is set on an odd half word address, the breakpoint is not taken if the instruction at the preceding even half word address is stalled due to an interlock.

Conditions

1. Debug is enabled (DBGEN is HIGH and Debug control register bit[5] is LOW)
2. 32-bit fetches in Thumb state are enabled (CFGTHUMB32 is HIGH)
3. Monitor mode debug is enabled (Debug control register bit[4] is HIGH)
4. A breakpoint is set on an odd halfword address (Address bit[1] = 1)
5. The instruction at the preceding even half word address is stalled due to an interlock

An example code sequence that causes this behavior is:

```
ADD R2, #4    ; Address = 0x100
LDR R0, [R1]  ; Address = 0x102
ADD R0, #1    ; Address = 0x104, Interlock due to data dependency on LDR
STR R0, [R2]  ; Address = 0x106, Breakpoint address
```

Implications

This errata only affects Monitor mode debug operations. The normal function of the core is not affected.

Workaround

To workaround this errata, use the BKPT instruction instead of setting breakpoints.

Errata - Category 3

318838: xTCMCANCEL not asserted when all data is read from the TCM write buffer

Status

Affects: product ARM966E-S.

Fault status: Cat 3, Present in: r2p0, Fixed in r2p1.

Description

This erratum affects designs using ITCMCANCEL and DTCMCANCEL to track whether the ARM966E-S has used the data that has been read.

Conditions

The fault occurs when all the data which is read comes from the TCM write buffer. For performance reasons, data is always read from memory, and combined with data in the TCM write buffer if necessary. If all of the data comes from the write buffer, the ARM966E-S should assert xTCMCANCEL to indicate that the data read has not been used.

Implications

It is not possible to rely on ARM966E-S asserting xTCMCANCEL if the data is read from the write buffer, and the data read from memory is discarded.

Workaround

Each TCM interface on the ARM966E-S can be switched to a mode where TCM accesses are executed in the order they were made. See section 2.3.8 of the ARM966E-S Rev.2 TRM for details. In this mode writes are always written to the TCM before the read is executed, and so data never comes back from the write buffer.

318839: HLOCK asserted unnecessarily on AHB

Status

Affects: product ARM966E-S.

Fault status: Cat 3, Present in: r2p0, Fixed in r2p1.

Description

This erratum affects designs using HLOCK to grant the ARM966E-S exclusive access to the AHB.

Conditions

The fault occurs when a write in the AHB write buffer is forced to flush by a SWP instruction. The write is unnecessarily locked with HLOCK.

Implications

This erratum may cause a small performance impact on systems which rely on HLOCK to only be asserted when necessary. There is no data corruption resulting from this erratum.

Workaround

There is no workaround.

412551: Coprocessor 15 instructions immediately following an MSR instruction may be executed with the wrong privilege level

Status

Affects: product ARM966E-S.

Fault status: Cat 3, Present in: r2p0,r2p1, Open.

Description

It is possible to change from a privileged mode to User mode by executing the MSR instruction. The subsequent 2 instructions will be present in the pipeline when the MSR instruction is executed. The MSR instruction does not cause the pipeline to be flushed. If the instruction immediately following the MSR instruction is a Coprocessor 15 operation it will be executed as if in a privileged mode rather than user mode.

Coprocessor 15 operations will not be executed when changing from User to privileged mode as the pipeline is flushed in these cases.

This erratum does not affect the return from exceptions as this usually involves an operation to change the program counter. Changing the program counter causes the pipeline to be flushed and so subsequent Coprocessor 15 operations will be executed with the correct privileges.

Conditions

For this behaviour to be exhibited, the following conditions must exist:

1. An MSR instruction is executed that changes from a privileged mode to User mode.
2. The instruction following the MSR instruction is a Coprocessor 15 operation.

Implications

The ARM Architecture Reference Manual recommends that an Instruction Memory Barrier (IMB) sequence is executed after using an MSR to change from a privileged mode to User mode.

If this is not done the erratum allows User mode access to all Coprocessor 15 operations including reading and writing of the System Control register. The code sequence that generates this behaviour would be very unusual. In most cases changing from User to privileged mode is done on return from a procedure, and this would use an instruction that would cause the pipeline to flush. It is most unusual for User code to immediately follow privileged code, where the last privileged instruction is to change to User mode.

Workaround

The erratum can be avoided by not placing coprocessor 15 instructions immediately after an MSR instruction. Placing a NOP instruction immediately after the MSR instruction will ensure that the correct privilege level is used for subsequent Coprocessor 15 operations.

```
MSR CPSR_c, Rm
NOP
<next instruction>
```

