ARM740T Rev 3 Errata List

CPU Cores Division

Document number:	ARM740T-PRDC-000594 1.0
Date of Issue:	10th August 2001
Author:	Phil Jones
Authorized by:	Keith Clarke

Abstract

This document describes the known errata in the ARM740T Revision 3 design.

Keywords

ARM7, ARM7TDMI, ARM740T, errata

This is a working document throughout the product lifecycle and, as such, the content may be modified as new information is uncovered.

Contents

1 ABOUT THIS DOCUMENT

1.1 Current History

Issue	Date	By	Change
1.0	6th August 2001	Phil Jones, CPU Product Manager	First issue.

1.2 References

Ref.	Document No	Author(s)	Title
1	ARM DDI 0008E	ARM	ARM740T Datasheet

1.3 Scope

This document describes the errata discovered in the implementation of the ARM740T Rev3, categorised by level of severity. Each description includes:

- where the implementation deviates from the specification
- the conditions under which erroneous behaviour occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a ‘workaround’ where possible
- the status of corrective action.

1.4 Terms and Abbreviations

This document uses the following terms and abbreviations.

Term	Meaning
Bounced Coprocessor Instruction	An invalid coprocessor instruction that results in the Undefined Instruction trap being taken.
Breakpoint	A debugging mechanism used to halt execution due to an **instruction fetch**. For the breakpoint to cause debug state entry, the instruction must reach the execution stage of the pipeline, but it will be prevented from executing. This enables a debugger to observe the state prior to that instruction's execution. A breakpoint can be made to occur by either appropriate triggering of the Embedded ICE watchpoint units or by asserting the **BREAKPT** signal during an instruction fetch.
DBGBREAK	EmbeddedICE breakpoint/watchpoint indicator. This signal identifies the current memory access with a debug condition. If the memory access is an instruction fetch then a **breakpoint** is indicated, otherwise a **watchpoint** is indicated.
DBGRQ	Debug request. An input used to signal the processor to enter debug state once the current executing instruction completes.
Debugger/Debugging Tool	A debugging system that includes a program used to detect, locate and correct software faults, together with custom hardware that supports software debugging.
Single-stepping	A debugging operation used to step through the flow of program execution, instruction by instruction. This can be implemented by using a single **watchpoint unit** configured to cause a **breakpoint** on the next instruction to be executed.
Watchpoint	A debugging mechanism used to halt execution due to a memory **data access**. Watchpoints cause debug state entry once the instruction causing the data access fully completes; this may include accepting state changes due to pending exceptions. Watchpoints enable a debugger to observe memory changes, such as updates to variables. A watchpoint can be made to occur by either appropriate triggering of the Embedded ICE watchpoint units or by asserting the **BREAKPT** signal during a data memory access.
Watchpoint Unit	Custom EmbeddedICE hardware capable of triggering a breakpoint or watchpoint when all its comparators match. Each watchpoint unit has several registers to configure the type of comparison desired, enabling matches against any value on the address bus, and/or the data bus and/or various bus control signals.

2 CATEGORISATION OF ERARATA

Errata recorded in this document are split into three groups:

Category 1 Features which are impossible to work around and severely restrict the use of the device in all or the majority of applications rendering the device unusable.

Category 2 Features which contravene the specified behaviour and may limit or severely impair the intended use of specified features but does not render the device unusable in all or the majority of applications.

Category 3 Features that were not the originally intended behaviour but should not cause any problems in applications.

2.1 Errata Summary

The errata associated with this product are categorised in the following way. Numbers in brackets after the errata description indicate the order in which the errata were found chronologically. The ARM7TDMI Rev.3a errata are included and have their own numbers independent from ARM740T specific errata.

Category 1	None
Category 2	Debug Request coincident with Aborting Store (ARM7TDMI - 1) Watchpoint and Prefetch Abort (ARM7TDMI - 2) Watchpoint coincident with Debug Request (ARM7TDMI - 3)
Category 3	None

3 CATEGORY 1 ERRATA

There are no errata in this group.

4 CATEGORY 2 ERRATA

4.1 Debug Request coincident with Aborting Store (ARM7TDMI - 1)

If **DBGRQ** or scan chain created debug request is asserted while an exception is being processed then the processor may not restart execution from the correct point after exiting debug state.

4.1.1 Conditions

A combination of two conditions is required to cause this erratum:

1) Debug Request is asserted and
2) A store instruction that causes a Data Abort on the final access is executing

If the above conditions are met then the debug entry mechanism fails to behave in the defined manner and the device may return from debug and execute from an incorrect address.

4.1.2 Implications

In this erratum, the Debug Request takes effect before the recognition of the abort. This may result in unreliable debug entry, the abort being missed and premature exit of debug state. Thus unintended or unpredictable device behaviour may result.

4.1.3 Workarounds

There is no practical workaround for this erratum. This is due to the difficulty in getting a debugging tool to recognise the symptoms of this erratum and take the appropriate corrective action. However the likelihood of failure with this mechanism is extremely low and the impact of failure is also low as it affects debugging operations only, therefore there is no plan to revise the design to resolve this erratum.

4.2 Watchpoint and Prefetch Abort (ARM7TDMI - 2)

If a watchpoint occurs followed by a prefetch aborted instruction then the processor may not restart execution from the correct point after exiting debug state.

4.2.1 Conditions

A combination of two conditions is required to cause this erratum:

1) An instruction that causes a watchpoint has been executed and
2) The second following instruction would Prefetch Abort is executed.

If the above conditions are met then the debug entry mechanism fails to behave in the defined manner and the device may return from debug and execute from the incorrect address.

4.2.2 Implications

In this erratum, the Prefetch Abort gets recognised prematurely, that is before debug entry occurs. This causes the PC to not advance by the expected amount as debug entry proceeds. Correspondingly, on return from debug the standard return address calculation produces an incorrect address and thus unintended or unpredictable device behaviour may result.

4.2.3 Workarounds

There is no practical workaround for this erratum. This is due to the difficulty in getting a debugging tool to recognise the symptoms of this erratum and take the appropriate corrective action. However the likelihood of failure with this mechanism is extremely low and the impact of failure is also low as it affects debugging operations only, therefore there is no plan to revise the design to resolve this erratum.

4.3 Watchpoint coincident with Debug Request (ARM7TDMI - 3)

If **DBGRQ** or the scan chain created debug request is asserted as debug entry due to a watchpoint is occurring then the processor may not restart execution from the correct point after exiting debug state.

4.3.1 Conditions

A combination of two conditions is required to cause this erratum:

1) Debug Request is asserted and
2) a load that causes a watchpoint is executing.

If the above conditions are met then the debug entry mechanism fails to behave in the defined manner and the device may return from debug and execute from the incorrect address.

4.3.2 Implications

In this erratum, the Debug Request takes affect before the recognition of the watchpoint. This may result in unreliable debug entry, the watchpoint being missed and premature exit of debug state. Thus unintended or unpredictable device behaviour may result.

4.3.3 Workarounds

There is no practical workaround for this erratum. This is due to the difficulty in getting a debugging tool to recognise the symptoms of this erratum and take the appropriate corrective action. However the likelihood of failure with this mechanism is extremely low and the impact of failure is also low as it affects debugging operations only, therefore there is no plan to revise the design to resolve this erratum.

5 CATEGORY 3 ERRATA

There are no errata in this group.

