ETM10 r0p0 Errata List

CPU Cores Division

Document number:	ETM10-PRDC-001383 Rev 4.0
Date of Issue:	11/11/2002
Author	John Hudson
Authorised by:	Ty Garibay

Abstract

This document describes the known errata in the ETM10 Rev0.

Keywords

Errata, bug, workaround, ETM10.

Contents

1 ABOUT THIS DOCUMENT 3

1.1 Change History 3

1.2 References 3

1.3 Scope 3

1.4 Terms and Abbreviations 3

2 CATEGORISATION OF ERRATA 4

2.1 Errata Summary Error! Bookmark not defined.

3 CATEGORY 1 ERRATA 5

4 CATEGORY 2 ERRATA 6

4.1 ETM10: 4-bit Demultiplexed Mode Requires External Logic for Compatibility with 8-bit and 16-bit Modes 6

5 CATEGORY 3 ERRATA 8

5.1 ETM10: Branch Address Output Twice Under Certain Conditions 8

5.2 ETM10: Incorrect Trace Data Outputted When Traced Thru POR 8

1 ABOUT THIS DOCUMENT

1.1 Change History

Issue	Date	By	Change
			Change History on Domino

1.2 References

This document refers to the following documents.

Ref.	Document No	Title
1	ARM IHI 0014	Embedded Trace Macrocell Specification

1.3 Scope

This document describes the errata discovered in the implementation of the ARM10200/ARM1020T/VFP10 Rev1, 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 ‘work-around’ where possible
- the status of corrective action.

1.4 Terms and Abbreviations

This document uses the following terms and abbreviations.

Term	Meaning

2 CATEGORISATION OF ERRATA

Errata recorded in this document are split into three groups:

Category 1 Features which are impossible to work around and severely restricts 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.

3 CATEGORY 1 ERRATA

There are no errata in this group.

4 CATEGORY 2 ERRATA

4.1 ETM10: Support for 4-bit Demultiplexed Mode Requires External Logic between the ETM10 and the Trace Capture Device connector

4.1.1 Summary

Additional on-chip logic is required to support the 4-bit demultiplexed mode

4.1.2 Description

The trace connector pin assignment for 4-bit demultiplexed mode is different from the 8-bit and 16-bit versions, to allow it to be supported by a single connector. These pin assignments are defined in the Embedded Trace Macrocell Specification [1].

Support for this 4-bit demultiplexed mode using a single connector is not provided by the ETM10. The ETM10 provides support only for the 8 and 16 bit demultiplexed modes, which require two trace connectors.

4.1.3 Conditions

This erratum will occur if 4-bit demultiplexed mode is selected.

4.1.4 Implications

4 bit demultiplexed trace is not possible with ETM10 unless the problem is fixed with additional logic between the ETM10 and the trace connector (see below), or the trace connectors are wired to only work in 4 bit Demultiplexed mode.

4.1.5 Workaround

The workaround is the addition of a small amount of logic between the ETM10 and the trace connector.

The normal connections for a single target connector are as follows:

CONNECTORA[38] = PIPESTATA[0]
CONNECTORA[36] = PIPESTATA[1]
CONNECTORA[34] = PIPESTATA[2]
CONNECTORA[32] = PIPESTATA[3]
CONNECTORA[30] = TRACEPKTA[0]
CONNECTORA[28] = TRACEPKTA[1]
CONNECTORA[26] = TRACEPKTA[2]
CONNECTORA[24] = TRACEPKTA[3]
CONNECTORA[22] = TRACEPKTA[4]
CONNECTORA[20] = TRACEPKTA[5]
CONNECTORA[18] = TRACEPKTA[6]
CONNECTORA[16] = TRACEPKTA[7]
...
CONNECTORA[37] = TRACEPKTA[8]
CONNECTORA[35] = TRACEPKTA[9]
CONNECTORA[33] = TRACEPKTA[10]
CONNECTORA[31] = TRACEPKTA[11]
CONNECTORA[29] = TRACEPKTA[12]
CONNECTORA[27] = TRACEPKTA[13]
CONNECTORA[25] = TRACEPKTA[14]
CONNECTORA[23] = TRACEPKTA[15]

Other connections are as defined in [1]. Connector B is connected to trace port B in a similar manner.

The these connections should be replaced by the following:

demul4 = ((PORTMODE[1:0] == 2'b10) && (PORTSIZE == 2'b00))

CONNECTORA[38] = PIPESTATA[0]
CONNECTORA[36] = PIPESTATA[1]
CONNECTORA[34] = PIPESTATA[2]
CONNECTORA[32] = PIPESTATA[3]
CONNECTORA[30] = TRACEPKTA[0]
CONNECTORA[28] = TRACEPKTA[1]
CONNECTORA[26] = TRACEPKTA[2]
CONNECTORA[24] = TRACEPKTA[3]
CONNECTORA[22] = TRACEPKTA[4]
CONNECTORA[20] = TRACEPKTA[5]
CONNECTORA[18] = TRACEPKTA[6]
CONNECTORA[16] = TRACEPKTA[7]
...
CONNECTORA[37] = (demul4 ? PIPESTATB[0] : TRACEPKTA[8])
CONNECTORA[35] = (demul4 ? PIPESTATB[1] : TRACEPKTA[9])
CONNECTORA[33] = (demul4 ? PIPESTATB[2] : TRACEPKTA[10])
CONNECTORA[31] = (demul4 ? PIPESTATB[3] : TRACEPKTA[11])
CONNECTORA[29] = (demul4 ? TRACEPKTB[0] : TRACEPKTA[12])
CONNECTORA[27] = (demul4 ? TRACEPKTB[1] : TRACEPKTA[13])
CONNECTORA[25] = (demul4 ? TRACEPKTB[2] : TRACEPKTA[14])
CONNECTORA[23] = (demul4 ? TRACEPKTB[3] : TRACEPKTA[15])

Other connections are unchanged. Connector B is unchanged, and can be directly connected to trace port B.

4.1.6 Corrective Action

None planned.

5 CATEGORY 3 ERRATA

5.1 ETM10: Branch Address Output Twice Under Certain Conditions

5.1.1 Summary

ETM10 may output a branch address twice.

5.1.2 Description

If the ETM10 is programmed such that it is turned ON in the middle of an LDM/STM this errata may occur. When ETM10 turns ON in the middle of LSM AND there is no executed instruction that cycle AND the first instruction afterwards happens to be a branch target, it is possible that the ETM10 may output the branch address twice.

5.1.3 Conditions

This erratum will occur in the middle of a LDM/STM, in which ETM is enabled, near branches.

5.1.4 Implications

The raw ETM trace data may contain the same branch happening twice. Fortunately, the ETM decompression software filters out this scenario, by design, wherein effectively masking the problem to the end programmer.

5.1.5 Workaround

The ETM de-compression software corrects the traced program output as viewed by the programmer.

5.1.6 Corrective Action

None planned.

5.2 ETM10: Incorrect Trace Data Outputted When Traced Thru POR

5.2.1 Summary

Erroneous Instruction Valid indication is sent to ETM by the core immediately following a power-on reset which then causes some incorrect trace data to appear if tracing through the power-on reset.

5.2.2 Description

When the ARM1020E core first comes out of reset, before the first instruction if fetched, a single Instruction Valid indication is sent to the ETM (ETMCORECTL[5]).

5.2.3 Conditions

For this errata to surface, the ETM must be programmed and enabled while the core is held in power-on reset. Tracing through a soft reset of the core will not result in this behaviour.

5.2.4 Implications

If tracing is enabled when the core first exits reset, the trace will begin with a branch to reset from an indeterminate address. The trace tools will show this as a branch from an unknown region. It is also possible that

the indeterministic address will cause a match of some sort in the Trigger logic. This could potentially result in erroneous assertion of TraceEnable, ViewData, and/or trigger an ETM event. However, it is highly unlikely for this to have any noticeable effect to the user.

5.2.5 Workaround

The recommended workaround is none since this this unlikely to surface in a typical debug session, even if tracing through reset is desired. However, it is possible to fix this problem through extra hardware by gating ETMCORECTL[5] with a resetable positive edge triggered flop that is initialized to zero and set by the first deassertion of ETMCORECTL[1]. This will effectively mask out the first incorrect assertion of ETMCORECTL[5]. This logic would need to be added to the ASIC in between the ARM1020E and the ETM10. Since all ARM1020/ETM10 paths currently contain no logic (besides the scan mux), the insertion of this gate would not result in any speedpath problems.

Example hdl code to correct problem.

```
reg ETMINSTVAL;
always @(CLK or RESET)
begin
  if(RESET)
    ETMINSTVAL <= 1'b0;
  else if(CLK)
    ETMINSTVAL = ETMINSTVAL & ETMCORECTL[1];
end
assign ETMCORECTL_NEW[5] = ETMINSTVAL & ETMCORECTL[5];
```

5.2.6 Corrective Action

None planned