ARM ETM
ETB11 (TM070)
Errata Notice

This document contains all errata known at the date of issue in releases up to and including revision r0p1 of Embedded Trace Buffer for ETM11 (ETB11)

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:errata@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:

- 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.
Implementation	Errata that are of particular interest to those implementing the product and that have no software implications.
System	Errata or possible issues that have system implications and therefore should be considered by system designers
Documentation	Errata or possible issues that have documentation implications.

Change Control

12 January 2009: Changes in Document revision 10.0

Page	Status	ID	Cat	Summary
13	New	641170	Cat 2	ETB11 AHB interface might operate incorrectly when BUSY transfers are used
19	Updated	424163	Sys	ETB11 AHB interface might respond incorrectly to IDLE transactions

15 May 2008: Changes in Document revision 9.0

Page	Status	ID	Cat	Summary
12	New	532313	Cat 2	ETB11 register accesses using JTAG interface might not operate correctly
14	New	521213	Cat 3	ETB11 fails to produce AHB error response
16	Updated	311060	Imp	Scan Enable incorrectly factored into ETB11 Trace RAM Write Enable
22	Updated	443372	Doc	RAM Output has to hold value when used in ETB11

29 Jun 2007: Changes in Document revision 8.0

Page	Status	ID	Cat	Summary
17	New	403350	Sys	ETB RAM location 0x0 might be corrupted when using the ETB RAM as system memory
19	New	424163	Sys	AHB interface might respond incorrectly to IDLE transactions
21	New	423312	Doc	ETB11 little endian device
22	New	443372	Doc	RAM Output has to hold value when used in ETB11

17 Sep 2003: Changes in Document revision 6.0

Page	Status	ID	Cat	Summary
16	New	311060	Imp	Scan Enable incorrectly factored into ETB11 Trace RAM Write Enable

19 Feb 2003: Changes in Document revision 5.0

Page	Status	ID	Cat	Summary
8	New	226809	Cat 1	Incorrect Data Value Read From ETB11 AHB Interface with Synchronous Operation

20 Dec 2002: Changes in Document revision 1.0

Page	Status	ID	Cat	Summary
9	New	232059	Cat 2	Potential Loss of Last Data Packet when Trace Capture is Disable

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	r0p1
311060	Imp	Scan Enable incorrectly factored into ETB11 Trace RAM Write Enable	X	X
403350	Sys	ETB RAM location 0x0 might be corrupted when using the ETB RAM as system memory	X	X
423312	Doc	ETB11 AHB interface is little endian	X	X
424163	Sys	ETB11 AHB interface might respond incorrectly to IDLE transactions	X	X
443372	Doc	RAM Output has to hold value when used in ETB11	X	X
226809	Cat 1	Incorrect Data Value Read From ETB11 AHB Interface with Synchronous Operation	X	
232059	Cat 2	Potential Loss of Last Data Packet when Trace Capture is Disabled	X	
532313	Cat 2	ETB11 register accesses using JTAG interface might not operate correctly	X	X
641170	Cat 2	ETB11 AHB interface might operate incorrectly when BUSY transfers are used	X	X
521213	Cat 3	ETB11 fails to produce AHB error response	X	X

Errata - Category 1

226809: Incorrect Data Value Read From ETB11 AHB Interface with Synchronous Operation

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Cat 1, Present in: r0p0, Fixed in r0p1.

Description

The ETB11 AHB Interface enables software access to the ETB11 Registers and RAM contents. In certain instances the ETB11 AHB slave interface may return incorrect data when in synchronous mode.

Conditions

This erratum occurs only if the ETB11 is configured for synchronous HCLK and CLK operation, i.e. with **SBYPASS** HIGH. The first read access to the ETB11 after an alternative AHB slave is accessed may result in the ETB11 responding with **HREADYMEM** HIGH before the requested data is available to the AHB bus.

Implications

A read access through the ETB11 AHB Interface results in incorrect data being read.

Workaround

SBYPASS may be set LOW so that the ETB11 operates asynchronously. Handshaking is performed between signals crossing multiple clock domains and this erratum does not occur. A consequence of this is that access times to the ETB11 are longer. This resolves this erratum and is acceptable to carry out in both synchronous and asynchronous designs.

If increased access times are unacceptable and **SBYPASS** is kept HIGH, then if the system contains a single AHB Master device (therefore guaranteeing that additional AHB transfers are not being performed at the same time), transfers must be repeated twice to ensure the correct data is accessed.

Errata - Category 2

232059: Potential Loss of Last Data Packet when Trace Capture is Disabled

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Cat 2, Present in: r0p0, Fixed in r0p1.

Description

A Data Formatter constructs ETB RAM data from trace that has been read in from the **TRACEOUTPUT** port. This is to allow multiple cycles of trace to be stored in a single word of ETB Trace RAM. If trace capture is disabled and the Data Formatter contains a partially built value, the Data Formatter inserts Trace Disabled (TD) packets until the data value is complete. Under certain configurations the last data word inserted with TD packets is not guaranteed to have been written to the RAM and is lost.

Conditions

Devices implementing ETM architecture versions ETMv3 and above, such as ETM10RV and ETM11RV are unaffected by this erratum.

The erratum occurs when the following four conditions are met:

- The ETM architecture version supplied to the ETB is:

- **PROTOCOL[1:0]** =b00 for ETMv1 architecture, or

- **PROTOCOL[1:0]** =b01 for ETMv2 architecture.

- The packet width **(TRACEPKT)** is set to:

- **PORTSIZE[2:0]** =b000 for **TRACEPKT** width = 4 bits, or

- **PORTSIZE[2:0]** =b001 for **TRACEPKT** width = 8 bits

- Software access to the ETB is disabled

- The software access pin **SWEN** for AHB access is set LOW, or

- Software access to ETB registers is disabled by setting the SoftwareCntl bit of the Control Register LOW.

- The ETB has not stopped capturing trace due to the trigger counter having reached 0. In other words, when reading the Status register (register 3), either:

- Triggered (bit 1) is LOW, or

- AcqComp (bit 2) is LOW.

If the ETB has stopped capturing trace due to the Trigger Counter Register decrementing to 0, then this erratum does not apply and Triggered and AcqComp are HIGH. If the ETB has not stopped capturing trace, or trace capture has been stopped for any other reason then this erratum may occur.

In summary, any data packets (a maximum of 2) remaining in the Data Formatter may be lost when:

(ETMv1 **OR** ETMv2 architecture) **AND**

(4-bit **OR** 8-bit trace packet width) **AND**

(Software Enable input = 0 **OR** SoftwareCntl = 0) **AND**

(AcqComp = 0 **OR** Triggered = 0).

This erratum does not apply if:

PROTOCOL[1:0] = b10 for the generic architecture used by ETMv3 devices, or

TRACEPKT width = 16 bits, or

SWEN and SoftwareCntl are HIGH, or

AcqComp and Triggered are HIGH.

Implications

Any data packets in the Data Formatter when trace capture is disabled (by setting the TraceCaptEn bit of the Control Register LOW) are not written to the ETB RAM and the data is lost:

- If **TRACEPKT** width = 4 bits a maximum of 2 trace packets may be lost.
- If **TRACEPKT** width = 8 bits a maximum of 1 trace packet may be lost.

Workaround

To guarantee that no data is lost under the conditions described in the implications, one of two methods can be used:

- If **SWEN** is HIGH then the ETB11 can be reprogrammed via JTAG to set SoftwareCntl HIGH at the same time as TraceCaptEn is set LOW.

- Select the INTEST instruction to put the ETB11 scan path into INTEST mode. SoftwareCntl is set LOW by doing this.
- Perform a write to the Control Register r8 with a value of 0x04.This sets SoftwareCntl HIGH and TraceCaptEn LOW.

- If **SWEN** is LOW, and it is not possible to change it to HIGH, then reprogram the ETM to produce additional trace data as described below.

Reprogramming the ETM to produce more trace data

The ETM should be configured to trace the minimum amount of additional data:

1. Configure and program the ETM to trace the desired data. Do not set the TraceCaptEn bit of the Control Register LOW at this point – Ensure that ETB trace capture is still enabled.
2. Reprogram the ETM as described in Table 1. This configures the ETM for:

- Cycle accurate instruction only tracing,
- of executed ARM state instructions,

- in the full address range,
- with ETM counter 1 set to start at a value of 2 and decrementing each cycle,
- with tracing disabled when the counter reaches 0, and never reloading.

This results in three additional instructions being traced.

Register	Value	Register Description	Configuration
0x00	Set bit [12] = 1	ETM control register	Enable cycle-accurate tracing, in addition to other control bits.
0x06	0x00000000	Trace start/stop resource control	No start/stop addresses
0x07	0x0000	TraceEnable control 2 register	No single address comparator include/exclude control
0x08	0x04040	TraceEnable event register	Counter 1 is non-zero
0x09	0x1000000	TraceEnable control 1 register	Exclude nothing
0x0C	0x0406F	ViewData event register	Always false, no data trace
0x10	0x00000000	Address comparator 1 value register	Address 0x00000000
0x11	0xFFFFFFFF	Address comparator 2 value register	Address 0xFFFFFFFF
0x20	0x019	Address comparator 1 access type register	ARM instruction, instruction execute
0x21	0x019	Address comparator 2 access type register	ARM instruction, instruction execute
0x50	0x0002	Counter 1 reload value register	Initial count = 0x2
0x54	0x00010	Counter 1 enable register	Address range comparator #
0x58	0x0406F	Counter 1 reload event	Always 0x0, never reload

Table 1: ETM programming to ensure ETB is fully drained of valid trace data

Set the TraceCaptEn bit of the Control Register LOW.

532313: ETB11 register accesses using JTAG interface might not operate correctly

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Cat 2, Present in: r0p0,r0p1, Open.

Description

The ETB supports RAM accesses and programming register accesses via an AHB interface or via the JTAG interface. Accesses over the AHB interface can be disabled by clearing the SoftwareCntl bit in the ETB Control register 8. This bit can only be modified by a JTAG access.

If the SoftwareCntl bit is set, writes to all ETB registers via the JTAG interface might be ignored except for writes to the SoftwareCntl bit. Any reads of ETB registers return an Unknown value.

Conditions

The following conditions must occur:

- The SoftwareCntl bit in the ETB Control register is set
- A read or write access is attempted to any ETB register using the JTAG interface

It should be noted that whenever INTEST is selected in the ETB JTAG TAP controller, the SoftwareCntl bit is automatically cleared. Therefore this erratum only occurs if debug tools deliberately set the SoftwareCntl bit while performing JTAG accesses.

Implications

Writes to all bits of all ETB registers might be ignored, except writes to the SoftwareCntl bit of the ETB Control register 8. This means the ETB might not be enabled correctly.

Reads of ETB registers return an Unknown value.

Workaround

This is a workaround for tools vendors.

Tools should ensure that when accessing the ETB via JTAG the SoftwareCntl bit is clear. If the SoftwareCntl bit is set then it must be cleared before accessing any ETB registers.

641170: ETB11 AHB interface might operate incorrectly when BUSY transfers are used

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Cat 2, Present in: r0p0,r0p1, Open.

Description

The ETB11 AHB Interface enables software access to the ETB11 registers and RAM contents. When BUSY transfers are used on the AHB interface, the ETB11 might operate incorrectly. It is dependent on the system whether BUSY transfers are generated.

Conditions

The following condition must occur:

- An AHB transaction occurs to the ETB11 where HTRANS indicates BUSY.

If only performing single transfers to the ETB11 an AHB master will typically not generate BUSY transfers, however this is dependent on the AHB master and the system interconnect.

Implications

Write accesses might be ignored and read accesses might return the wrong value. This erratum affects accesses to the ETB11 RAM and control registers.

This erratum is likely to affect the usability of the ETB11 RAM as system memory.

The AHB protocol is not violated.

Workaround

This is a workaround for system implementors:

The HTRANS[0] input to the ETB11 must be forced LOW. This has the effect of converting all BUSY transactions into IDLE transactions and all SEQ transactions into NSEQ transactions. This does not affect the performance of the ETB11.

This is a workaround for programmers using software to access the ETB11:

If performing multiple transfer instructions to access the ETB11 such as LDM or STM instructions, these should be replaced by equivalent single transfer instructions. Typically an AHB master will not generate BUSY transfers for these instructions, however this must be checked with the vendor of your AHB master and system interconnect.

Errata - Category 3

521213: ETB11 fails to produce AHB error response

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Cat 3, Present in: r0p0,r0p1, Open.

Description

The ETB11 supports RAM accesses and programming register accesses via an AHB interface or via the JTAG interface. Accesses over the AHB interface can be disabled by clearing the SoftwareCntl bit in the ETB Control register 8. This bit can only be cleared by a JTAG access, or whenever the INTEST instruction is selected using the JTAG interface. The reset status of the SoftwareCntl bit is HIGH. AHB accesses can also be disabled by driving the SWEN input to the ETB11 LOW.

When AHB accesses are disabled the ETB11 should respond with an ERROR response over the AHB interface. This erratum might cause the ETB11 to respond with an OK response.

Conditions

The following conditions must occur:

- At least one of the following must occur:
- The SWEN input to the ETB11 is LOW
- The SoftwareCntl bit in the ETB11 Control register is LOW
- An AHB access is performed to the ETB

Implications

If the AHB access was a read, then the value returned is unknown. If the AHB access was a write, the value written will not update the ETB11 register or RAM contents.

The software which initiated the AHB access will not know that the access failed and therefore might behave in an unexpected manner.

It is expected that this situation will not occur during normal usage of the ETB11. There are two main uses of the ETB11, for trace capture and as system memory. When used as system memory, it is not expected that a debugger will interact with the ETB11 and as such the SoftwareCntl bit will always be HIGH and this erratum does not occur. When used for trace capture, the ETB11 is usually only accessed by either the JTAG interface or the AHB interface and as such this erratum is not expected to occur.

Workaround

There is no workaround for this erratum.

Users should be aware that simultaneous accesses to the ETB from on-chip software and an external debugger should be avoided.

Errata - Implementation

311060: Scan Enable incorrectly factored into ETB11 Trace RAM Write Enable

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Imp, Present in: r0p0,r0p1, Open.

Description

When the SE (Scan Enable) input to ETB11 is high, the Write Enable to the ETB11 Trace RAM is always active.

Conditions

This only affects the operation of ETB11 when in manufacturing test mode. This does not affect the functional operation of ETB11.

Implications

This can result in random data being written into the RAM and means that the RAM interface cannot be fully tested with full-sequential ATPG patterns.

Investigation using TetraMAX (version 2002.05-3) to generate full-sequential stuck-at ATPG patterns shows the difference in fault coverage with the Write Enable (WE) to the RAM forced low or forced high is negligible (99.77% with WE forced high and 99.69% with WE forced low).

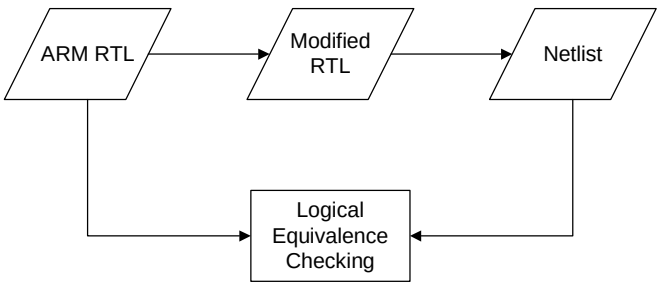
Workaround

The WE term in Etb11SRAMIntf.v (line 153) can be changed to:

assign WE = WEbus & {4{~SE}};

This forces WE inactive when SE is high. This modification does affect the functional operation of ETB11.

You must synthesize the modified ETB11 and verify the netlist is functionally equivalent to the original ETB11 Rev.0p1 RTL.



Errata - System

403350: ETB RAM location 0x0 might be corrupted when using the ETB RAM as system memory

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Sys, Present in: r0p0,r0p1, Open.

Description

The AMBA AHB specifications allow a master to insert BUSY cycles during a burst transfer. An attempt to write to the ETB11 via AHB while HTRANS is set to BUSY for at least 5 cycles will corrupt location 0x0 in the ETB11 RAM.

Conditions

The following conditions must occur:

- SBYPASS set
- A burst transfer of at least two items with 5 BUSY cycles between items
- HWRITE set for the entirety of the burst transfer
- HSELMEM set for the entirety of the burst transfer

Implications

The AHB master driving the ETB11 might assume that the IDLE transaction will always result in a zero wait-state response. This erratum might cause the AHB master to drive a new non-IDLE transaction on the AHB interface while HREADYMEM is low and therefore causing the ETB11 to miss the new transaction.

The non-IDLE transaction might have incorrect data returned in the case of a read transaction, or a write transaction might be performed incorrectly in the ETB11.

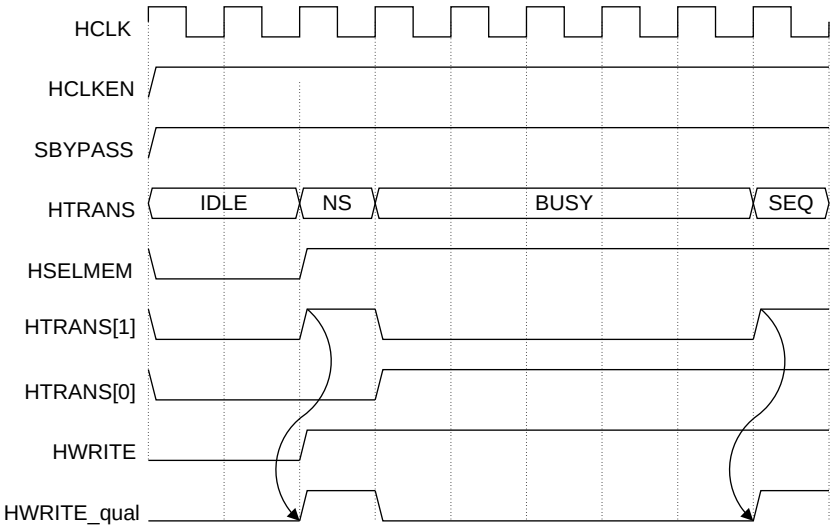
This erratum does not affect usage when only accessing via JTAG.

Workaround

This is a workaround for system implementors.

The HWRITE signal must be qualified by HTRANS[1] before being connected to the ETB11.

```
assign HWRITE_qual    = HWRITE & HTRANS[1]
```



424163: ETB11 AHB interface might respond incorrectly to IDLE transactions

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Sys, Present in: r0p0,r0p1, Open.

Description

The AMBA AHB specification requires that a slave must respond to an IDLE transaction with a zero wait-state OK response. The ETB11 might not always respond with a zero wait-state OK response to IDLE transactions.

Conditions

The following conditions must occur:

- One of HSELMEM or HSELREG is HIGH
- An IDLE transaction is indicated on HTRANS

The HREADYMEM output from the ETB11 will unexpectedly be driven low during the data phase of the IDLE transaction, dependent on a certain internal state of the ETB11.

Implications

The AHB master driving the ETB11 might assume that the IDLE transaction will always result in a zero wait-state response. This erratum might cause the AHB master to drive a new non-IDLE transaction on the AHB interface while HREADYMEM is low and therefore causing the ETB11 to miss the new transaction.

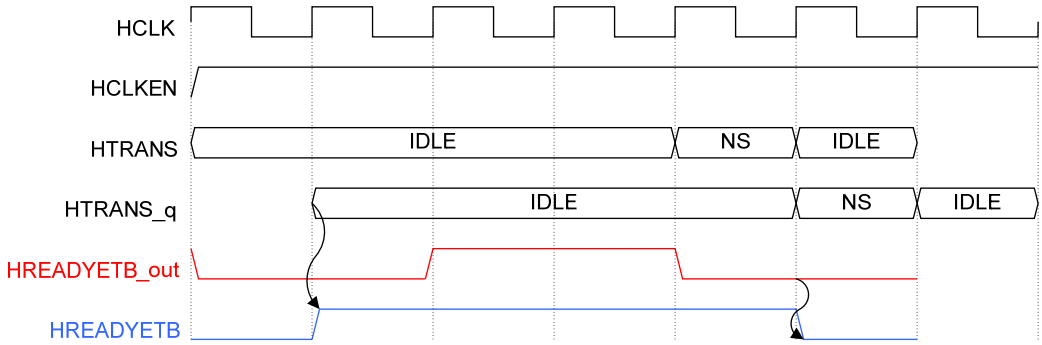
The non-IDLE transaction might have incorrect data returned in the case of a read transaction, or a write transaction might be performed incorrectly in the ETB11.

This erratum does not affect usage when only accessing via JTAG.

Workaround

This is a workaround for system implementors.

The HREADYETB output from the ETB11 must be qualified based on whether an IDLE transaction is in progress. It is important that this is done during the data phase of the transfer (second cycle) and therefore a registered version of HTRANS should be used. While this registered version of HTRANS is b00 HREADYETB must be high.



This can be achieved with the following logic applied to HREADYETB output from the ETB11 creating a new HREADY output.

```
module ETB11SetHready (/*AUTOARG*/
  // Inputs
  HCLK, HCLKEN, HRESETn, HTRANS, HREADYETB_out, HREADYBUS,   // Outputs
  HREADYETB
  );
  input       HCLK;          // AHB Clock
  input       HCLKEN;        // AHB Clock Enable
  input       HRESETn;       // AHB Reset
  input [1:0] HTRANS;        // AHB Trans
  input       HREADYETB_out; // HREADY from ETB11 (slave)
  input       HREADYBUS;     // HREADY from Master
  output      HREADYETB;     // New ETB11 HREADY
  // wire declaration
  wire        htrans_en;     // Enable when to sample HTRANS
  reg [1:0]   htrans_q;      // register version of HTRANS
  wire        d_phase_idle;  // during a data phase of a transfer HTRANS is IDLE
  // ------------------------------------------------------------------------
  // Main code
  // ------------------------------------------------------------------------
  // Clock gating term to select HTRANS
  assign      htrans_en = HCLKEN & HREADYBUS;

  always @(posedge HCLK or negedge HRESETn)
    if (!HRESETn)
      htrans_q <= 2'b00;
    else if (htrans_en)
      htrans_q <= HTRANS;
  // We use the register version of HTRANS because we are interested in the data
  // phase of the transfer.
  assign      d_phase_idle = (htrans_q == 2'b00);
  // If trans is IDLE the HREADY signal will always be high
  assign      HREADYETB = d_phase_idle | HREADYETB_out;
endmodule
```

Errata - Documentation

423312: ETB11 AHB interface is little endian

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Doc, Present in: r0p0,r0p1, Open.

Description

The AHB interface on the ETB11 is a little-endian interface. This means that if the ETB11 is used in a big-endian memory system and byte or half-word accesses are performed to the ETB11 RAM using the AHB interface then the bytes in the transfer must be rotated. This rotation should either be performed with on-chip logic, or by the software which is accessing the ETB11 RAM.

Implications

If the byte rotation is not performed then the data in the RAM might be incorrectly interpreted.

Workaround

System implementors must be aware that the ETB11 is a little endian device and must ensure that the correct byte rotations are performed for byte and half-word accesses.

443372: RAM Output has to hold value when used in ETB11

Status

Affects: product Embedded Trace Buffer for ETM11 (ETB11).

Fault status: Doc, Present in: r0p0,r0p1, Open.

Description

The ETB11 Implementation Guide (ARM DII 0067C) must have the following additional description of the RAM used with the ETB11:

The RAM implemented with the ETB11 must hold read data on the Q output between RAM accesses. If the RAM does not currently perform this holding function, a latch must be implemented to hold the data.

The Q output is ignored by the ETB11 on write accesses, so the Q data does not need to be held for writes.