ARM ETM
CoreSight ™ ETM11™ (TM920)
Errata Notice

This document contains all errata known at the date of issue in CoreSight ETM11 releases up to and including revision r1p1 of CoreSight ETM11.

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 four 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.

Change Control

16 May 2007: Changes in Document revision 10.0

Page	Status	ID	Cat	Summary
86	New	435327	Imp	PCLKEN input to ETM11CSSingle might cause timing closure failures

23 Mar 2007: Changes in Document revision 9.0

Page	Status	ID	Cat	Summary
47	Updated	408879	Cat 2	Periodic Synchronisation might not occur
50	New	425203	Cat 2	Trigger might not occur
52	New	426912	Cat 2	Incorrect byte lanes selected in Big-Endian mode
68	Updated	409113	Cat 3	ETMJTAGPORT behaves incorrectly when simultaneous JTAG and APB transactions occur
85	New	426940	Cat 3	Imprecise data abort causes comparator to fire incorrectly
81	New	424963	Cat 3	ETMJTAGPORT prevents System APB accesses when nTRST is low
82	New	425205	Cat 3	Multiple trigger requests might occur
83	New	426367	Cat 3	Address comparator compares against incorrect Context-ID
84	New	426463	Cat 3	Address range comparator behaves incorrectly on unaligned transfer as ProgBit is cleared

14 Dec 2006: Changes in Document revision 8.0

Page	Status	ID	Cat	Summary
47	New	408879	Cat 2	Periodic Synchronisation might not occur
68	New	409113	Cat 2	ETMJTAGPORT behaves incorrectly when simultaneous JTAG and APB transactions occur
49	New	412415	Cat 2	Precisely aborted unaligned transfers might cause unpredictable data address comparator behavior
70	New	410117	Cat 3	Address Range comparator matches incorrectly on 64-bit transfer
71	New	411183	Cat 3	Address Range comparator with Exact Match bit set fails to match after a coprocessor access
67	Updated	407022	Cat 3	Extra Store-Failed packet produced when using ViewData to filter store-exclusive data transfers
72	New	411191	Cat 3	TraceEnable Start/Stop resource stops early
73	New	411334	Cat 3	Address Range comparator is not cleared by a coprocessor access
74	New	412312	Cat 3	Data address comparator might match incorrectly around folded branch instruction
76	New	412335	Cat 3	ViewData might behave unpredictably if sensitive to a Single Address Comparator with the Exact Match bit set
77	New	413518	Cat 3	Address Range comparator with exact match bit set might not match around unaligned transfers
79	New	413723	Cat 3	PCLKENDBG not used in claim tag register
80	New	416413	Cat 3	Data value trace might be missing around unaligned stores

09 Oct 2006: Changes in Document revision 7.0

Page	Status	ID	Cat	Summary
20	Updated	350999	Cat 2	Setting or clearing the programming bit might generate corrupt trace
22	Updated	351599	Cat 2	Context ID in I-Sync packet might be incorrect
23	Updated	353052	Cat 2	Instruction Address in I-Sync packet might be incorrect following STREXD instruction
25	Updated	354104	Cat 2	ATCLK registers might return incorrect value
26	Updated	355901	Cat 2	Register reads via ETM JTAG Port might return incorrect values
27	Updated	363115	Cat 2	Wait For Interrupt causes incorrect trace
29	Updated	364367	Cat 2	Failed store-exclusive transfers cause data comparators to behave incorrectly
30	Updated	369866	Cat 2	Tracing the upper 32-bits of a failed STREXD instruction might cause corrupt trace
31	Updated	369872	Cat 2	A-Sync packet might not be output when trace is enabled
33	Updated	369963	Cat 2	ETM FIFO might not empty when setting the programming bit
35	Updated	374140	Cat 2	De-assertion of NIDEN and DBGEN might cause incorrect trace
36	Updated	392132	Cat 2	Normal CPRT transfer might be traced as a Context-ID packet
39	Updated	395608	Cat 2	Accesses to ATCLK registers can cause APB interface to lockup
41	Updated	397332	Cat 2	Context ID might be updated incorrectly when tracing is prohibited
43	Updated	397333	Cat 2	Context ID might be incorrectly updated when ETM is powered down
44	Updated	398636	Cat 2	AFREADYM signal might not be asserted
46	Updated	398884	Cat 2	Out-of-order data might be incorrect
60	Updated	364310	Cat 3	Authentication Status register value is incorrect
61	Updated	374728	Cat 3	Lock Access can be modified when PADDRDBG31 is HIGH
59	Updated	349882	Cat 3	Writes to counter value register might cause counter event to fire immediately
62	Updated	398993	Cat 3	Cycle Accurate Count inaccurate on exit from overflow
63	New	400435	Cat 3	ViewData programmed to be sensitive to a non-data instruction might behave incorrectly
65	New	404712	Cat 3	Incorrect Context ID and security state might be reported at trace turn-on
67	New	407022	Cat 3	Extra Store-Failed packet produced when using ViewData to filter store-exclusive data transfers

31 Jul 2006: Changes in Document revision 6.0

Page	Status	ID	Cat	Summary
36	Updated	392132	Cat 2	Normal CPRT transfer might be traced as a Context-ID packet
39	New	395608	Cat 2	Accesses to ATCLK registers can cause APB interface to lockup
41	New	397332	Cat 2	Context ID might be updated incorrectly when tracing is prohibited

44	New	398636	Cat 2	AFREADYM signal might not be asserted
43	New	397333	Cat 2	Context ID might be incorrectly updated when ETM is powered down
46	New	398884	Cat 2	Out-of-order data might be incorrect
62	New	398993	Cat 3	Cycle Accurate Count inaccurate on exit from overflow

22 Jun 2006: Changes in Document revision 5.0

Page	Status	ID	Cat	Summary
23	Updated	353052	Cat 2	Instruction Address in I-Sync packet might be incorrect following STREXD instruction
30	New	369866	Cat 2	Tracing the upper 32-bits of a failed STREXD instruction might cause corrupt trace
31	New	369872	Cat 2	A-Sync packet might not be output when trace is enabled
33	New	369963	Cat 2	ETM FIFO might not empty when setting the programming bit
35	New	374140	Cat 2	De-assertion of NIDEN and DBGEN might cause incorrect trace
36	New	392132	Cat 2	Normal CPRT transfer might be traced as a Context-ID packet
61	New	374728	Cat 3	Lock Access can be modified when PADDRDBG31 is HIGH

28 Oct 2005: Changes in Document revision 4.0

Page	Status	ID	Cat	Summary
18	Updated	345646	Cat 2	Imprecisely aborted unaligned data accesses may be traced incorrectly
19	Updated	346298	Cat 2	Extra instruction traced when a stalled coprocessor data transfer is followed by branch fold onto another data transfer
20	Updated	350999	Cat 2	Setting or clearing the programming bit might generate corrupt trace
27	New	363115	Cat 2	Wait For Interrupt causes incorrect trace
60	New	364310	Cat 3	Authentication Status register value is incorrect
29	New	364367	Cat 2	Failed store-exclusive transfers cause data comparators to behave incorrectly

22 Aug 2005: Changes in Document revision 3.0

Page	Status	ID	Cat	Summary
54	Updated	345320	Cat 3	Branch Output bit can be set, but feature not supported
55	Updated	345449	Cat 3	Unaligned big-endian (BE-32) data transfer addresses inferred incorrectly by ETM
56	Updated	345456	Cat 3	Data missed from trace when next to branch-phantom at start of include region with exact-match bit set in data-only mode
18	Updated	345646	Cat 2	Imprecisely aborted unaligned data accesses may be traced incorrectly
19	Updated	346298	Cat 2	Additional instruction traced when a stalled data transfer is followed by branch fold onto another data transfer
57	Updated	348902	Cat 3	Interrupted stores might be unexpectedly traced

59	Updated	349882	Cat 3	Writes to counter value register might cause counter event to fire immediately
20	New	350999	Cat 2	Disabling trace by setting the programming bit might generate corrupt trace
22	New	351599	Cat 2	Context ID in I-Sync packet might be incorrect
23	New	353052	Cat 2	Instruction Address in I-Sync packet might be incorrect following STRDEX instruction
25	New	354104	Cat 2	ATCLK registers might return incorrect value
26	New	355901	Cat 2	Register reads via ETM JTAG Port might return incorrect values

12 May 2005: Changes in Document revision 2.0

Page	Status	ID	Cat	Summary
54	New	345320	Cat 3	Branch Output bit can be set, but feature not supported
55	New	345449	Cat 3	Unaligned big-endian (BE-32) data transfer addresses inferred incorrectly by ETM
56	New	345456	Cat 3	Data missed from trace when next to branch-phantom at start of include region with exact-match bit set in data-only mode
18	New	345646	Cat 2	Imprecisely aborted unaligned data accesses may be traced incorrectly
19	New	346298	Cat 2	Additional instruction traced when a stalled data transfer is followed by branch fold onto another data transfer
57	New	348902	Cat 3	Interrupted stores might be unexpectedly traced
59	New	349882	Cat 3	Writes to counter value register might cause counter event to fire immediately

19 Jul 2004: Changes in Document revision 1.0

No Errata in this document revision

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	r0p2	r1p0	r1p1
345646	Cat 2	Imprecisely aborted unaligned data accesses may be traced incorrectly	X				
346298	Cat 2	Extra instruction traced when a stalled coprocessor data transfer is followed by branch fold onto another data transfer	X				
350999	Cat 2	Setting or clearing the programming bit might generate corrupt trace	X	X			
351599	Cat 2	Context ID in I-Sync packet might be incorrect	X	X			
353052	Cat 2	Instruction Address in I-Sync packet might be incorrect following STREXD instruction	X	X			
354104	Cat 2	ATCLK registers might return incorrect value	X	X			
355901	Cat 2	Register reads via ETM JTAG Port might return incorrect values	X	X			
363115	Cat 2	Wait For Interrupt causes incorrect trace	X	X			
364367	Cat 2	Failed store-exclusive transfers cause data comparators to behave incorrectly	X	X			
369866	Cat 2	Tracing the upper 32-bits of a failed STREXD instruction might cause corrupt trace	X	X			
369872	Cat 2	A-Sync packet might not be output when trace is enabled	X	X			
369963	Cat 2	ETM FIFO might not empty when setting the programming bit	X	X			
374140	Cat 2	De-assertion of NIDEN and DBGEN might cause incorrect trace	X	X			
392132	Cat 2	Normal CPRT transfer might be traced as a Context-ID packet	X	X			
395608	Cat 2	Accesses to ATCLK registers can cause APB interface to lockup	X	X			
397332	Cat 2	Context ID might be updated incorrectly when tracing is prohibited	X	X			
397333	Cat 2	Context ID might be incorrectly updated when ETM is powered down	X	X			
398636	Cat 2	AFREADYM signal might not be asserted	X	X			
398884	Cat 2	Out-of-order data might be incorrect	X	X			
408879	Cat 2	Periodic Synchronisation might not occur			X	X	
412415	Cat 2	Precisely aborted unaligned transfers might cause unpredictable data address comparator behavior	X	X	X		
425203	Cat 2	Trigger might not occur	X	X			

ID	Cat	Summary of Erratum	r0p0	r0p1	r0p2	r1p0	r1p1
426912	Cat 2	Incorrect byte lanes selected in Big-Endian mode	X	X	X	X	
345320	Cat 3	Branch Output bit can be set, but feature not supported	X				
345449	Cat 3	Unaligned big-endian (BE-32) data transfer addresses inferred incorrectly by ETM	X				
345456	Cat 3	Data missed from trace when next to branch-phantom at start of include region with exact-match bit set in data-only mode	X				
348902	Cat 3	Interrupted stores might be unexpectedly traced	X				
349882	Cat 3	Writes to counter value register might cause counter event to fire immediately	X	X			
364310	Cat 3	Authentication Status register value is incorrect	X	X			
374728	Cat 3	Lock Access can be modified when PADDRDBG31 is HIGH	X	X			
398993	Cat 3	Cycle Accurate Count inaccurate on exit from overflow	X	X			
400435	Cat 3	ViewData programmed to be sensitive to a non-data instruction might behave incorrectly	X	X			
404712	Cat 3	Incorrect Context ID and security state might be reported at trace turn-on	X	X	X	X	X
407022	Cat 3	Extra Store-Failed packet produced when using ViewData to filter store-exclusive data transfers			X		
409113	Cat 3	ETMJTAGPORT behaves incorrectly when simultaneous JTAG and APB transactions occur	X	X	X	X	
410117	Cat 3	Address Range comparator matches incorrectly on 64-bit transfer	X	X	X		
411183	Cat 3	Address Range comparator with Exact Match bit set fails to match after a coprocessor access	X	X	X		
411191	Cat 3	TraceEnable Start/Stop resource stops early			X		
411334	Cat 3	Address Range comparator is not cleared by a coprocessor access	X	X	X		
412312	Cat 3	Data address comparator might match incorrectly around folded branch instruction	X	X	X		
412335	Cat 3	ViewData might behave unpredictably if sensitive to a Single Address Comparator with the Exact Match bit set	X	X	X		
413518	Cat 3	Address Range comparator with exact match bit set might not match around unaligned transfers	X	X	X	X	
413723	Cat 3	PCLKENDBG not used in claim tag register	X	X	X		
416413	Cat 3	Data value trace might be missing around unaligned stores				X	
424963	Cat 3	ETMJTAGPORT prevents System APB accesses when nTRST is low	X				
425205	Cat 3	Multiple trigger requests might occur	X	X			
426367	Cat 3	Address comparator compares against incorrect Context-ID	X	X	X	X	

ID	Cat	Summary of Erratum	r0p0	r0p1	r0p2	r1p0	r1p1
426463	Cat 3	Address range comparator behaves incorrectly on unaligned transfer as ProgBit is cleared	X	X	X	X	
426940	Cat 3	Imprecise data abort causes comparator to fire incorrectly	X	X	X	X	
435327	Imp	PCLKEN input to ETM11CSSingle might cause timing closure failures	X	X	X	X	

Affected Processors Table

The CoreSight ETM11 is compatible with multiple processors. The following table indicates which errata are possible within the ETM when connected to each processor.

A cell shown thus **X** indicates that the defect can only affect the ETM when connected to the processor shown at the top of that column.

ID	Cat	Summary of Erratum	ARM1136J(F)-S r0	ARM1136J(F)-S r1	ARM1156T2(F)-S	ARM1176JZ(F)-S	ARM11 MPCore
345646	Cat 2	Imprecisely aborted unaligned data accesses may be traced incorrectly				X	
346298	Cat 2	Extra instruction traced when a stalled coprocessor data transfer is followed by branch fold onto another data transfer				X	
350999	Cat 2	Setting or clearing the programming bit might generate corrupt trace	X	X	X	X	
351599	Cat 2	Context ID in I-Sync packet might be incorrect	X	X	X	X	
353052	Cat 2	Instruction Address in I-Sync packet might be incorrect following STREXD instruction		X		X	
354104	Cat 2	ATCLK registers might return incorrect value	X	X	X	X	
355901	Cat 2	Register reads via ETM JTAG Port might return incorrect values	X	X	X	X	
363115	Cat 2	Wait For Interrupt causes incorrect trace	X	X	X	X	
364367	Cat 2	Failed store-exclusive transfers cause data comparators to behave incorrectly	X	X	X	X	
369866	Cat 2	Tracing the upper 32-bits of a failed STREXD instruction might cause corrupt trace		X		X	
369872	Cat 2	A-Sync packet might not be output when trace is enabled	X	X	X	X	
369963	Cat 2	ETM FIFO might not empty when setting the programming bit	X	X	X	X	
374140	Cat 2	De-assertion of NIDEN and DBGEN might cause incorrect trace	X	X	X	X	
392132	Cat 2	Normal CPRT transfer might be traced as a Context-ID packet	X	X	X	X	
395608	Cat 2	Accesses to ATCLK registers can cause APB interface to lockup	X	X	X	X	
397332	Cat 2	Context ID might be updated incorrectly when tracing is prohibited				X	

ID	Cat	Summary of Erratum	ARM1136J(F)-S r0	ARM1136J(F)-S r1	ARM1156T2(F)-S	ARM1176JZ(F)-S	ARM11 MPCore
397333	Cat 2	Context ID might be incorrectly updated when ETM is powered down				X	
398636	Cat 2	AFREADYM signal might not be asserted	X	X	X	X	
398884	Cat 2	Out-of-order data might be incorrect	X	X	X	X	
408879	Cat 2	Periodic Synchronisation might not occur	X	X	X	X	X
409113	Cat 2	ETMJTAGPORT behaves incorrectly when simultaneous JTAG and APB transactions occur	X	X	X	X	X
412415	Cat 2	Precisely aborted unaligned transfers might cause unpredictable data address comparator behavior	X	X	X	X	
425203	Cat 2	Trigger might not occur	X	X	X	X	
426912	Cat 2	Incorrect byte lanes selected in Big-Endian mode	X	X	X	X	
345320	Cat 3	Branch Output bit can be set, but feature not supported				X	
345449	Cat 3	Unaligned big-endian (BE-32) data transfer addresses inferred incorrectly by ETM				X	
345456	Cat 3	Data missed from trace when next to branch-phantom at start of include region with exact-match bit set in data-only mode				X	
348902	Cat 3	Interrupted stores might be unexpectedly traced				X	
349882	Cat 3	Writes to counter value register might cause counter event to fire immediately	X	X	X	X	
364310	Cat 3	Authentication Status register value is incorrect				X	
374728	Cat 3	Lock Access can be modified when PADDRDBG31 is HIGH	X	X	X	X	
398993	Cat 3	Cycle Accurate Count inaccurate on exit from overflow	X	X	X	X	
400435	Cat 3	ViewData programmed to be sensitive to a non-data instruction might behave incorrectly	X	X	X	X	
404712	Cat 3	Incorrect Context ID and security state might be reported at trace turn-on				X	
407022	Cat 3	Extra Store-Failed packet produced when using ViewData to filter store-exclusive data transfers		X		X	
410117	Cat 3	Address Range comparator matches incorrectly on 64-bit transfer	X	X	X	X	
411183	Cat 3	Address Range comparator with Exact Match bit set fails to match after a coprocessor access	X	X	X	X	
411191	Cat 3	TraceEnable Start/Stop resource stops early	X	X	X	X	
411334	Cat 3	Address Range comparator is not cleared by a coprocessor access	X	X	X	X	

ID	Cat	Summary of Erratum	ARM1136J(F)-S r0	ARM1136J(F)-S r1	ARM1156T2(F)-S	ARM1176JZ(F)-S	ARM11 MPCore
412312	Cat 3	Data address comparator might match incorrectly around folded branch instruction	X		X	X	
412335	Cat 3	ViewData might behave unpredictably if sensitive to a Single Address Comparator with the Exact Match bit set	X	X	X	X	
413518	Cat 3	Address Range comparator with exact match bit set might not match around unaligned transfers	X	X	X	X	X
413723	Cat 3	PCLKENDBG not used in claim tag register	X	X	X	X	
416413	Cat 3	Data value trace might be missing around unaligned stores					X
424963	Cat 3	ETMJTAGPORT prevents System APB accesses when nTRST is low	X	X	X	X	
425205	Cat 3	Multiple trigger requests might occur	X	X	X	X	
426367	Cat 3	Address comparator compares against incorrect Context-ID				X	
426463	Cat 3	Address range comparator behaves incorrectly on unaligned transfer as ProgBit is cleared	X	X	X	X	X
426940	Cat 3	Imprecise data abort causes comparator to fire incorrectly	X	X	X	X	
435327	Imp	PCLKEN input to ETM11CSSingle might cause timing problems	X	X	X	X	X

Errata - Category 1

There are no Errata in this Category.

Errata - Category 2

345646: Imprecisely aborted unaligned data accesses may be traced incorrectly

Status

Affects: product CoreSight ETM11.

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

Description

If the first half of an unaligned data access is imprecisely aborted but the second is successful, the data is traced incorrectly and the exception is not reported.

Conditions

1. The first part of an unaligned data access is imprecisely aborted
2. The second part of the unaligned data access is successful
3. Data trace is enabled

Implications

Imprecisely aborted unaligned data accesses might not be traced correctly. The data value might be incorrect and the imprecise data abort exception might not be indicated in the trace.

Workaround

There is no workaround.

346298: Extra instruction traced when a stalled coprocessor data transfer is followed by branch fold onto another data transfer

Status

Affects: product CoreSight ETM11.

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

Description

When a coprocessor data transfer instruction is followed by a branch folded onto another data transfer, a spurious additional instruction may be traced.

Conditions

1. Instruction trace is enabled
2. A coprocessor data transfer is followed by a branch folded onto another data transfer

Implications

Trace is incorrect until the next indirect branch or periodic I-sync.

Workaround

There is no workaround. Instruction trace synchronisation is regained at the next periodic I-sync packet. For this reason, increasing the frequency of periodic I-sync packets is recommended.

350999: Setting or clearing the programming bit might generate corrupt trace

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When setting or clearing the programming bit in the ETM Control register (0x000), the ETM can enter a state where corrupt trace data is output. The programming bit is used in the ETM to disable all tracing functionality, allowing the ETM registers to be programmed.

There are two sets of conditions where this erratum can occur, when setting or clearing the programming bit.

Conditions when setting the programming bit

1. TraceEnable is HIGH and the ETM is tracing
2. The ETM programming bit is set by writing to the ETM Control register, 0x000

If the erratum occurs under this set of conditions, up to 144 byes of incorrect trace are output.

Conditions when clearing the programming bit

1. The ETM is configured to generate trace
2. The ETM programming bit is cleared by writing to the ETM Control register, 0x000

If the erratum occurs under this set of conditions, any trace data output is corrupted.

Implications

If this erratum occurs, corrupt trace data is output. Trace decompressors might lose synchronisation when decompressing this trace.

Workaround

This workaround is for tool vendors.

When enabling the CoreSight ETM11, the ETM must be enabled but configured not to generate any trace data. It must then be disabled, and re-enabled to generate trace in the desired configuration. The sequence of operations is as follows:

1. Power up ETM (if not already performed) using the ETM Control register 0x000
2. Program up all ETM registers except the TraceEnable Event register 0x008
3. Program the TraceEnable event register 0x008 with 0x0000406F, which disables trace generation
4. Clear the programming bit in the ETM Control register 0x000
5. Wait for the programming bit to be cleared in the ETM status register 0x004
6. Set the programming bit in the ETM Control register 0x000
7. Wait for the programming bit to be set in the ETM status register 0x004

8. Configure the TraceEnable Event register 0x008 as originally desired
9. Clear the programming bit in the ETM Control register 0x000
10. Wait for the programming bit to be cleared in the ETM status register 0x004

351599: Context ID in I-Sync packet might be incorrect

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When tracing is turned on, the CoreSight ETM11 generates an I-Sync packet to indicate the current state of the processor, including the current Context-ID. If tracing is turned on one cycle after the data transfer for a Context-ID changing instruction, the Context-ID in the I-Sync packet might be the old Context-ID. The new value of the Context-ID is not traced.

This erratum only affects non-periodic I-Sync packets. Periodic I-Sync packets are not affected.

Conditions

1. Context-ID tracing is enabled
2. Tracing is enabled one cycle after the data transfer for a Context-ID changing instruction

Implications

When tracing is turned on, the current Context-ID might be incorrect. Trace decompression tools might rely on the Context-ID to determine the opcodes of instructions and as such they might interpret the trace incorrectly until the next periodic I-Sync packet, when the Context-ID is traced again.

Workaround

There is no workaround for this erratum, however if tracing is enabled for a sufficiently long period, the erratum is detectable. Trace decompression tools must identify when the Context-ID in a periodic I-Sync does not match the expected current Context-ID obtained from the last non-periodic I-Sync packet. If a mismatch is detected, the Context-ID used since the non-periodic I-Sync must be discarded. For this reason, increasing the frequency of periodic I-Sync packets is recommended when Context-ID tracing is enabled, since periodic I-Sync packets are not affected by this erratum.

353052: Instruction Address in I-Sync packet might be incorrect following STREXD instruction

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When tracing is turned on, the CoreSight ETM11 generates an I-Sync packet to indicate the current state of the processor, which includes the address of the next instruction to be xecuted. Turning on trace just after the data for a failed STREXD instruction might cause the address in the I-Sync packet to be incorrect.

Conditions

Table 1 gives a detailed cycle by cycle breakdown of the conditions for the erratum to occur.

1. The CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor or ARM1136J(F)-S rev 1 processor
2. The processor is in high-performance mode
3. Instruction tracing is enabled
4. Data Address tracing and Data value tracing is enabled
5. A STREXD instruction is executed and is traced
6. Out-of-order placeholder trace packets are generated for the STREXD instruction
7. Tracing is disabled (TraceEnable is de-asserted)
8. Another data instruction is executed
9. Tracing is re-enabled the cycle after the out-of-order data packet returns for the STREXD instruction, where the data is indicated to have failed the exclusive operation

Table 1: Detailed erratum breakdown

Cycle	Actual Execution	Traced Execution	Comments
1	0x1000: STREXD instruction	0x1000: STREXD instruction	Traced correctly
1	First STREXD transfer out-of-order placeholder	First STREXD transfer out-of-order placeholder	Traced correctly
1	Second STREXD transfer out-of-order placeholder	Second STREXD transfer out-of-order placeholder	Traced correctly
2	0x1004: Second data instruction		Not traced because TraceEnable is LOW
3	Out-of-order data for first STREXDtransfer		Not traced because TraceEnable is LOW

3	Out-of-order data for second STREXDtransfer		Not traced because TraceEnable is LOW
3		I-Sync packet indicating address 0x1050	Tracing is enabled on the next cycle, forcing an I-Sync packet on this cycle. The address is incorrect and should be 0x1004.
4	Data for instruction at 0x1004	Data for instruction at 0x1004	Traced correctly

Implications

When this erratum occurs, the least significant byte of the instruction address is always 0x50. Since the instruction address is incorrect, instruction trace is incorrect until the next indirect branch or periodic I-Sync packet.

Workaround

There is no workaround for this erratum. Instruction trace synchronisation is regained at the next indirect branch or periodic I-sync packet. For this reason, increasing the frequency of periodic I-sync packets is recommended.

354104: ATCLK registers might return incorrect value

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

The readable registers in the ETM11CS programmer's model which are clocked using ATCLK might return 32'h00000000 instead of the correct value. The affected registers are:

- CoreSight Trace ID, register 0x080 (offset 0x200)
- ITTRIGGERACK, register 0x3B9 (offset 0xEE4)
- ITATBCTR2, register 0x3BC (offset 0xEF0)

When any two of the above registers are read in close succession, the second register read might return 32'h00000000. The minimum time between reads to avoid this erratum is dependent on the frequency of ATCLK and PCLKDBG.

Conditions

1. One of the aforementioned registers is read
2. A second register on the aforementioned list is read

Implications

The read data from the second register is incorrect.

Workaround

This is a workaround for tool vendors.

The CoreSight ETM11 is implemented in 2 types of system, based on the method of accessing the ETM registers:

1. The ETM is accessed via the ETM JTAG Port. The erratum does not occur in these systems since the ATCLK registers are not accessible via the ETM JTAG Port.
2. The ETM is accessed via the Debug-APB interface, controlled by a Debug Access Port (DAP). In these systems, ATCLK is always equal to or faster than PCLKDBG and this workaround must be applied.

When reading any of the aforementioned registers, four register reads must be made from a register in the PCLKDBG domain prior to reading the ATCLK domain register. This ensures the read data for the ATCLK register is ready to be passed out over the APB interface.

It is recommended that the PCLKDBG domain register read is the Component ID0 (register 0x3FC, offset 0xFF0). The four reads will each return a data value of 0x0000000D, which must be ignored.

355901: Register reads via ETM JTAG Port might return incorrect values

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

This erratum only affects CoreSight ETM11 when used with the ETM JTAG Port, as connected in ETM11CSSingle. If the ETM JTAG Port is not used, this erratum does not occur.

When reading a register via the ETM JTAG Port, 2 shifts of 40 bits must take place in the Shift-DR state of the JTAG TAP state machine. The first 40 bits indicate a read of a register is required and the second 40 bits are used to scan out the result of the read. Between the shifts of 40 bits, the JTAG State machine must go through the Update-DR state and return to Shift-DR. Whilst returning to Shift-DR, if the JTAG state machine does not pass through Run-Test/Idle, the read data returned in the second 40 bit shift will be incorrect.

Conditions

1. The CoreSight ETM11 is used with the ETM JTAG Port
2. A register read is performed where the JTAG TAP state machine does not pass through Run-Test/Idle between the read command being scanned in and the read data being scanned out.

Implications

The data returned when reading ETM registers via JTAG is incorrect.

Workaround

This workaround is for tool vendors.

When reading ETM registers, the JTAG TAP state machine must pass through the Run-Test/Idle state when transitioning from Update-DR to Shift-DR. This increases the number of TCK cycles taken to move between Update-DR and Shift-DR by one cycle.

363115: Wait For Interrupt causes incorrect trace

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When an ARM11 processor executes a Wait For Interrupt (WFI) instruction, the CoreSight ETM11 stops tracing, empties the trace data FIFO and enters an idle state. This erratum might cause one or both of the following situations to occur when the WFI instruction is executed

1. The CoreSight ETM11 outputs 4 bytes of incorrect trace
2. The CoreSight ETM11 outputs incorrect trace when the processor restarts, until the ETM is reprogrammed

Conditions

1. The CoreSight ETM11 is enabled, but might not be tracing
2. The ARM11 processor executes a WFI instruction

Implications

This erratum causes incorrect trace to be output and prevents accurate trace capture after a WFI instruction, for example when the ARM11 processor enters dormant mode.

For situation 1 in the erratum description, up to 4 random bytes of trace appear in the trace stream. Packet boundary synchronisation is lost, and all trace captured since the WFI instruction must be discarded up to the next A-Sync packet.

For situation 2 in the erratum description, incorrect trace is output until the ETM programming bit is set. All trace captured since the WFI instruction must be discarded.

Workaround

There are two workarounds for users with code which contains Wait For Interrupt instructions.

It should be noted that if the user does not have modification capabilities to routines which execute WFI instructions there is no workaround for this erratum.

The first workaround is to remove any WFI instructions from the code. This prevents the processor from entering dormant or shutdown mode. When applying this workaround, users must take care to ensure the code which uses the WFI instruction behaves correctly without the WFI instruction.

The second workaround can only be applied if the system has software access to the CoreSight ETM11.

This workaround requires code to be written to disable the CoreSight ETM11 whilst the WFI instruction is executed. The code consists of two parts, one part to disable the ETM before the WFI instruction executes and a second part to re-enable the ETM when the processor restarts after WFI. Before applying the workaround, the ETM Control register (register 0x000) must be read and if bit [10] is HIGH the workaround must not be applied because the ETM is not enabled.

The following pseudo-code must be executed before the WFI instruction to disable the ETM:

```
Read ETM Control register (register 0x000) into register Rn
```

```
    Set bit [10] of Rn
    Write Rn to ETM Control register (register 0x000)
    Save ETM Status register (register 0x004) to memory
    Save ETM Counter 1 value register (register 0x05C) to memory
    Save ETM Counter 2 value register (register 0x05D) to memory
    Save ETM Sequencer state register (register 0x067) to memory
label1
    Read ETM status register (register 0x004) into register Rn
    Loop to "label1" if bit [1] of Rn is clear
The following pseudo-code must be executed on restart after WFI to re-enable the ETM
correctly:
    Read ETM TraceEnable Event register (0x008) into register Rm
    Write 0x0000406f to ETM TraceEnable Event register (0x008)
    Read ETM Control register (register 0x000) into register Rn
    Clear bit [10] of Rn
    Write Rn to ETM Control register (register 0x000)
label2
    Read ETM status register (register 0x004) into register Rn
    Loop to "label2" if bit [1] of Rn is set
    Read ETM Control register (register 0x000) into register Rn
    Set bit [10] of Rn
    Write Rn to ETM Control register (register 0x000)
label3
    Read ETM status register (register 0x004) into register Rn
    Loop to "label3" if bit [1] of Rn is clear
    Write Rm to ETM TraceEnable Event register (register 0x008)
    Restore ETM Status register (register 0x004) from memory
    Restore ETM Counter 1 value register (register 0x05C) from memory
    Restore ETM Counter 2 value register (register 0x05D) from memory
    Restore ETM Sequencer state register (register 0x067) from memory
    Read ETM Control register (register 0x000) into register Rn
    Clear bit [10] of Rn
    Write Rn to ETM Control register (register 0x000)
label4
    Read ETM status register (register 0x004) into register Rn
    Loop to "label4" if bit [1] of Rn is set
```

This workaround is invasive to the routines used to execute a WFI instruction, and prevents tracing of the WFI instruction and any code executed immediately on restart after WFI, until the ETM re-enabling code is completed.

364367: Failed store-exclusive transfers cause data comparators to behave incorrectly

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When a store-exclusive data transfer fails, the CoreSight ETM11 should ignore the data value of the transfer and any comparators configured for data address comparisons should behave as defined in the ETM Architecture specification.

This erratum has two different effects, based on how the address comparator is configured to use a data value comparator, as controlled by bits [6:5] of the Address Access Type register (0x20 to 0x27):

- Data value comparisons are enabled

This erratum causes the CoreSight ETM11 to incorrectly use the data value presented by the ARM11 processor during a failed store-exclusive transfer in the data value comparison. This result is used when determining the final result of the address comparator.

- Data value comparisons are disabled

This erratum causes the CoreSight ETM11 to ignore whether the store-exclusive transfer failed or succeeded. If the exact match bit (bit [7] of the Address Access Type register) is clear, this erratum has no effect on the result of the comparison. If the exact match bit is set, the comparator might unexpectedly match.

Conditions

1. The CoreSight ETM11 is configured to monitor data addresses which are accessed by store-exclusive instructions
2. The ARM11 processor executes a failed store-exclusive instruction which accesses the monitored addresses

Implications

If a failed store-exclusive data transfer occurs to an address which an ETM address comparator has been configured to monitor, the behaviour of the address comparator is unpredictable. Any ETM resources which have been configured to be sensitive to this address comparator might not behave as intended.

Workaround

There is no workaround for this erratum.

369866: Tracing the upper 32-bits of a failed STREXD instruction might cause corrupt trace

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When tracing a failed store-exclusive data transfer, a data packet is generated by the CoreSight ETM11, followed by a Store-Failed packet. For STREXD instructions which fail, 2 data packets are generated and each one is followed by a Store-Failed packet.

Using the ViewData resource in the CoreSight ETM11 allows zero, one or both data packets of the failed STREXD data transfers to be traced. If ViewData is set to only trace the second data transfer, corrupt trace might be generated by the CoreSight ETM11.

Conditions

1. The CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor or an ARM1136J(F)-S r1 processor
2. The CoreSight ETM11 is configured to trace data addresses, data values or both
3. ViewData is configured to use one or more address comparators as include or exclude regions. This might mean that single addresses are excluded or included, or ranges of addresses are included or excluded.
4. The ARM11 processor executes a failed STREXD instruction. The above ViewData configuration causes the upper word access to be traced and the lower word access not to be traced.

For example, ViewData might be programmed to include Single Address Comparator 1. Single Address Comparator 1 is programmed to match when a data transfer to address 0x1004 is performed. If a failed STREXD instruction accesses the 32-bit words at 0x1000 and 0x1004 then the ViewData configuration would cause only the second transfer at 0x1004 to be traced, thereby causing this erratum.

Implications

If this erratum occurs, corrupt trace data is output. Trace decompressors might lose synchronisation when decompressing this trace.

Workaround

There is no workaround for this erratum.

Trace synchronisation is regained at the next A-Sync packet. Therefore, increasing the synchronisation frequency is recommended to reduce the impact of this erratum.

369872: A-Sync packet might not be output when trace is enabled

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

An alignment synchronisation (A-Sync) packet is required every time the ETM programming bit is cleared, to ensure trace decompression tools can determine the packet boundary of the trace protocol. A-Sync packets are also output periodically in the trace stream.

When the ETM programming bit is cleared, this erratum might cause an A-Sync packet to be missing from the trace output.

Conditions

1. The programming bit in the ETM control register is cleared.
2. The ATCLK input to the CoreSight ETM11 is running much slower than the CLK input.

Implications

If this erratum occurs, packet boundary alignment might not be possible when tracing is enabled. This is important if trace is stored in a circular buffer, because trace synchronisation might not be possible.

Workaround

In ETMs not subject to this erratum, trace decompression tools perform the following sequence in order to synchronise with the instruction trace:

1. Search from the beginning of the captured trace until an A-Sync packet is found.
2. Search from the A-Sync packet until an I-Sync packet is found.
3. Start instruction trace decompression.

In addition, a data synchronisation packet must be found before data trace decompression can begin. This aspect is not discussed further.

If the trace capture device has not wrapped around (that is, no part of the trace stream has been overwritten), then the trace stream should start as follows when this erratum does not occur:

1. A-Sync packet.
2. Non-periodic I-Sync packet.
3. Other trace packets.

When this erratum occurs, the A-Sync packet is missing and the trace decompressor can work around the erratum in the following way:

- If it is known that the trace capture device has not wrapped around, then the decompressor can omit the search for an A-Sync packet. The trace will begin on a packet boundary, and an A-Sync is not required.
- If it is known that the trace capture device has wrapped around, then this erratum has no implications and the normal decompression sequence must be followed.

- If it is not known whether the trace capture device has wrapped around, then the decompressor can look at the beginning of the trace stream to determine if it starts with a non-periodic I-Sync packet. If the beginning of the trace can be decompressed as an I-Sync packet, then decompression should continue from this point.

A side effect of the third option is that it is possible, under exceptional circumstances, for the beginning of the captured trace stream to not be an I-Sync packet but to be a valid encoding for an I-Sync packet. An example would be if the trace capture device has wrapped around and the captured trace starts in the middle of the trace packet. If this occurs then a decompression error will occur when the decompressor attempts to decompress a subsequent packet. The decompressor will then be forced to search for an A-Sync and I-Sync after this point, as normal.

If it is not possible for the trace decompression tools to implement this workaround, then you can reduce its impact by increasing the synchronisation frequency. To do this, write a lower value to the Synchronisation Frequency Register (ETM register number 0x78). This will reduce the amount of discarded trace.

369963: ETM FIFO might not empty when setting the programming bit

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When setting the programming bit (bit [10] of the ETM Control register, 0x000), the CoreSight ETM11 stops tracing and empties its internal trace data FIFO. When the FIFO is empty, bit [1] of the ETM Status register (0x004) is set to 1'b1 to indicate the ETM has reached a stable state. This erratum might cause bit [1] of the ETM Status register to never be set to 1'b1.

Two sets of conditions might cause this erratum, described in Conditions 1 and Conditions 2.

This erratum does not occur if the processor is halted before the ETM programming bit is set.

Conditions 1

The following sequence must occur in order.

1. The CoreSight ETM11 is enabled and TraceEnable is active
2. The ARM11 processor executes an indirect branch instruction
3. The programming bit in the CoreSight ETM11 is set
4. The ARM11 processor executes the instruction at the destination of the indirect branch

Conditions 2

The following sequence must occur in order.

1. The CoreSight ETM11 is enabled and TraceEnable is active
2. The ARM11 processor executes a data transfer instruction which performs one or more data transfers
3. An Out-of-Order data placeholder packet is traced for at least one of the transfers
4. The programming bit in the CoreSight ETM11 is set
5. An Out-of-Order data packet is generated for the earlier Out-of-Order placeholder packet

Implications

The ETM Architecture specification describes the process by which programming the ETM registers must be performed. Part of this process describes what to perform when setting the programming bit and includes instructions to read the ETM Status register until bit [1] is set. If this erratum occurs, the programming process cannot complete because bit [1] of the ETM Status register is never set.

Workaround

This is a workaround for tools vendors.

When setting the programming bit, the ETM Status register must be read until bit [1] is set. If this does not occur within a reasonable number of iterations (it is recommended that you use a minimum of 200 iterations), perform the following steps:

1. Save the value of the TraceEnable Event register (register 0x008, offset 0x020)
2. Write 0x0000406F to the TraceEnable Event register, to disable trace generation
3. Clear the programming bit
4. Set the programming bit
5. Read the ETM Status register until bit [1] is set

If bit [1] of the ETM Status Register still does not return set, double the iteration limit and try again. Repeat the above steps until the status register reads as set.

374140: De-assertion of NIDEN and DBGEN might cause incorrect trace

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

Two inputs signals, NIDEN and DBGEN, are provided on CoreSight ETM11 as global control signals to allow the CoreSight ETM11 to operate.

If both of these signals are LOW, then the CoreSight ETM11 should stop tracing and output all trace currently in the FIFO. If either of the signals is subsequently driven HIGH, the CoreSight ETM11 should restart tracing at the following instruction boundary.

If this erratum occurs, the ETM FIFO does not empty when these signals are driven LOW. When one of the signals is driven HIGH, the data remaining in the FIFO is output, but the ETM might not restart tracing correctly and might output incorrect trace. Packet boundary synchronisation is maintained.

Conditions

The following operations must occur in the sequence defined:

1. The CoreSight ETM11 is enabled and generating trace
2. NIDEN and DBGEN are driven LOW
3. NIDEN or DBGEN is driven HIGH

Implications

The trace data is incorrect until the next I-Sync packet or indirect branch packet.

It is not expected that the signals NIDEN and DBGEN will be dynamically changed during tracing since the normal usage model for these signals is to permanently disable tracing on a device. If these signals are not dynamically changed during tracing, this erratum does not occur.

Workaround

It is recommended that NIDEN and DBGEN are not dynamically changed during tracing. If they are dynamically changed during tracing then there is no workaround for this erratum.

Trace synchronisation can be regained at the next indirect branch packet or I-Sync packet.

392132: Normal CPRT transfer might be traced as a Context-ID packet

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

Normally, when MCR or MRC instructions are executed the data transferred between the processor and the coprocessor is traced as normal data packets by the ETM if the MonitorCPRT bit (bit [1] of the ETM Control Register) is set. If the ContextIDSize bits (bits [15:14] of the ETM Control register) are set to anything but b00, then the data for Context-ID updating MCR instructions is traced as a Context-ID packet, independently of the setting of the MonitorCPRT bit.

When this erratum occurs, a normal MCR/MRC data transfer is incorrectly interpreted as a Context-ID updating data transfer and the ETM generates a Context-ID packet. This is incorrect since it indicates to the tools that the Context-ID has been updated when it was not updated.

One or more of the conditions described below in Conditions 1 to Conditions 4 might cause this erratum to occur.

Conditions 1

The following sequence must occur in order for this erratum to occur:

1. The ETM is configured to trace Context-ID (bits [15:14] of the ETM Control register are not b00).
2. A Context-ID updating MCR instruction is fetched by the ARM11 processor.
3. An interrupt or imprecise data abort occurs, which cancels the MCR instruction. The MCR instruction is presented to the ETM, but is subsequently indicated as cancelled on the ETM interface.
4. Zero or more instructions might be executed.
5. An MCR or MRC instruction is executed which is not a Context-ID changing instruction while tracing is enabled.

Tracing does not need to be enabled for conditions 1 to 4, but must be enabled for condition 5.

Conditions 2

This sequence can only occur when the CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor.

The following sequence must occur in order for this erratum to occur:

1. The ETM is configured to trace Context-ID (bits [15:14] of the ETM Control register are not b00)
2. The ARM processor enters a region where trace is prohibited
3. The first instruction which is executed when tracing is prohibited is a Context-ID updating MCR instruction
4. The ARM processor leaves the region where trace is prohibited
5. An MCR or MRC instruction is executed which is not a Context-ID changing instruction while tracing is enabled.

Tracing does not need to be enabled for conditions 1 to 4, but must be enabled for condition 5.

Entry to the prohibited trace region described in conditions 2 and 4 can be caused by a number of separate conditions:

- The SPNIDEN input signal to the ARM1176JZ(F)-S processor is de-asserted whilst the processor is executing code in a Secure Privileged mode.
- The ARM1176JZ(F)-S processor enters a Secure Privileged mode whilst the SPNIDEN input signal is de-asserted.

Conditions 3

The following sequence must occur in order for this erratum to occur:

1. The ETM is configured to trace Context-ID (bits [15:14] of the ETM Control register are not b00)
2. A Context-ID updating MCR instruction is executed by the ARM11 processor
3. The ETM is powered down by setting the ETMPWRDOWN bit in the ETM Control Register (bit [0] of register 0x000)
4. The ETM is powered up by clearing the ETMPWRDOWN bit in the ETM Control register
5. An MCR or MRC instruction is executed which is not a Context-ID changing instruction while tracing is enabled.

Tracing does not need to be enabled for conditions 1 to 4, but must be enabled for condition 5.

Conditions 4

The following sequence must occur in order for this erratum to occur:

1. The ETM is configured to trace Context-ID (bits [15:14] of the ETM Control register are not b00)
2. A Context-ID updating MCR instruction is executed by the ARM11 processor
3. Both the NIDEN and DBGEN input signals to the CoreSight ETM11 are de-asserted
4. One or both of the NIDEN and DBGEN input signals to the CoreSight ETM11 are asserted
5. An MCR or MRC instruction is executed which is not a Context-ID changing instruction while tracing is enabled.

Tracing does not need to be enabled for conditions 1 to 4, but must be enabled for condition 5.

Implications

The ETM incorrectly indicates the Context-ID has been updated and trace decompression tools might use this to select an incorrect memory image to use for decompression of the trace stream. Tools might also be unable to find the correct memory image. As such the trace might be interpreted incorrectly by trace decompression tools.

The Context-ID will be correctly output at the next periodic or non-periodic I-Sync packet, allowing correct decompression from the I-Sync packet.

This erratum does not affect the Context-ID comparators or any events based upon the Context-ID comparators.

Workaround

This is a workaround for tools vendors when instruction tracing is enabled. There is no workaround if instruction tracing is disabled.

When a Context-ID packet is decompressed, the MCR instruction which caused it will also have been traced. If the most recent MCR/MRC instruction is not a Context-ID updating MCR instruction, then this erratum has occurred and the Context-ID packet must be treated as a normal data packet.

When tracing an ARM1176JZ-S or ARM1176JZF-S processor, Context-ID packets are output when the security level changes to indicate that the currently active Context-ID has changed. These Context-ID packets are not associated with any MCR instruction. When detecting this erratum, if the next packet is a branch packet indicating a security level change, then this Context-ID packet has been correctly output.

395608: Accesses to ATCLK registers can cause APB interface to lockup

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

The CoreSight ETM11 performs internal clock gating to decrease power consumption. Many of the internal registers are clock gated when any of the following conditions exist:

1. The ETMPWRDOWN (bit [0] of the ETM Control register, 0x000) is HIGH
2. The input signals NIDEN and DBGEN are both LOW

When the internal registers are clock gated, accesses to the programmer's model registers in the ATCLK domain cause the output signal PREADYDBG to remain LOW, thereby causing the APB interface to lock up and preventing further accesses to the ETM and other devices on the same APB bus until the ETM is reset.

Conditions

1. The ETMPWRDOWN (bit [0] of the ETM Control register, 0x000) is HIGH, OR the input signals NIDEN and DBGEN are both LOW
2. One of the following registers is accessed:
   a. CoreSight Trace ID register (register 0x080, offset 0x200)
   b. Integration Register ITATBCTR0 (register 0x3BE, offset 0xEF8)
   c. Integration Register ITATBCTR1 (register 0x3BD, offset 0xEF4)
   d. Integration Register ITATBCTR2 (register 0x3BC, offset 0xEF0)
   e. Integration Register ITATBDATA0 (register 0x3BB, offset 0xEEC)
   f. Integration Register ITTRIGGERREQ (register 0x3BA, offset 0xEE8)
   g. Integration Register ITTRIGGERACK (register 0x3B9, offset 0xEE4)

These registers cannot be accessed via JTAG when the ETM11CS is used in the ETM11CSSingle configuration. As such this erratum cannot occur if the only supported access mechanism is via JTAG to ETM11CSSingle.

Implications

When this erratum occurs, the PREADYDBG output from the CoreSight ETM11 is driven low until PRESETDBGn is asserted LOW. This means the APB bus is locked up and no further accesses can be made to the CoreSight ETM11 or any other peripherals on the same APB bus.

This erratum does not affect JTAG accesses to the CoreSight ETM11 when using the ETM JTAG PORT in ETM11CSSingle, since the ATCLK registers cannot be accessed in this configuration.

This erratum is highly unlikely to occur, since debug tools must always power up the ETM before accessing the affected registers. If NIDEN and DBGEN are both LOW, tools must inspect the Authentication Status register (register 0x3EE, offset 0xFB8) to determine if non-invasive debug is enabled before accessing the affected registers. The only other likely reason to access these registers is by faulty software, for example where a corrupted pointer causes the ETM registers to be accessed by accident.

Workaround

This is a workaround for tools vendors.

Tools should ensure that the proper mechanisms are used to detect if the ETM is powered up and non-invasive debug is enabled before accessing the affected registers. The ETMPWRDOWN bit (bit [0] of the ETM Control register, register 0x000) must be LOW and the Authentication Status register (register 0x3EE, offset 0xFB8) must indicate non-invasive debug is enabled.

There is no workaround when faulty software accidentally accesses the ETM.

397332: Context ID might be updated incorrectly when tracing is prohibited

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When the CoreSight ETM11 is connected to an ARM1176JZ(F)-S, the CoreSight ETM11 maintains a shadow copy of both the Secure and Non-Secure Context ID registers. Writes to the User Read Only Thread and Process ID register might be interpreted by the CoreSight ETM11 as a Context ID change. The instruction to update the User Read Only Thread and Process ID register is:

MCR p15, 0, <Rd>, c13, c0, 3

This erratum only occurs when the processor is in Secure Privileged state and tracing is prohibited. Tracing is prohibited in Secure Privileged modes when the SPNIDEN input signal is LOW.

Conditions

1. The CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor
2. The processor is in a Secure Privileged state
3. The SPNIDEN input signal to the ARM1176JZ(F)-S is LOW
4. The User Read Only Thread and Process ID register is written using the instruction above

When the above conditions occur, one of the ETM's copies of the Context ID register is updated, depending on the value of the NS bit in the ARM1176JZ(F)-S processor. If the NS bit is CLEAR, the ETM's copy of the Secure Context ID is incorrectly updated. If the NS bit is SET, the ETM's copy of the Non-Secure Context ID is incorrectly updated.

The User Read Only Thread and Process ID register is part of the ARMv6 Advanced OS extensions and is used by Operating Systems to store Thread and Process IDs.

Implications

The ETM uses its copies of the Context ID registers for two purposes:

- The Context ID is output in the trace stream when it is changed or when the security level is changed or as part of periodic synchronisation
- The current Context ID is compared against a user defined Context ID value using the Context ID comparators and the result is used as an event resource throughout the ETM

Both of these purposes will use the incorrect Context ID.

If the Non-Secure Context ID was updated as described in the conditions above, then when the processor next enters Non-Secure state, the incorrect Context ID is output. Additionally, the Context ID comparator will compare against an incorrect Context ID value and any resources in the ETM which are dependent on this comparator will behave incorrectly. The incorrect Non-Secure Context ID is used until the Non-Secure Context ID is updated with a new value.

If the Secure Context ID was updated as described in the conditions above, then this erratum is not visible immediately since tracing is prohibited. If the SPNIDEN signal is subsequently asserted HIGH the incorrect

Secure Context ID will be used. The incorrect Secure Context ID is used until the Secure Context ID is updated with a new value.

Workaround

This erratum only occurs if the User Read Only Thread and Process ID register is written to. There is no workaround for this erratum.

397333: Context ID might be incorrectly updated when ETM is powered down

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

The ETM has an internal clock gating mechanism which is used to conserve power when the ETM is not in use. When the ETMPWRDOWN bit (bit [0] of the ETM Control register 0x000) is set, the ETM is disabled and most of the ETM is clock gated. Some logic in the ETM remains clocked to enable the ETM to track the current Context ID so that the current Context ID can be broadcast when the ETM is enabled. The ETM maintains a shadow copy of both the Secure and Non-Secure Context IDs.

This erratum causes the incorrect Context ID to be updated when the ETMPWRDOWN bit is cleared.

Conditions

The following conditions must occur in order:

1. The ETM is connected to an ARM1176JZ(F)-S processor
2. The ETM is powered down, the ETMPWRDOWN bit is b1
3. The Context ID is updated by writing to the relevant System Control Coprocessor Register
4. The ETM is powered up, the ETMPWRDOWN bit is b0

If the Non-Secure Context ID is updated in condition 3 above, then when the ETMPWRDOWN bit is cleared the ETM's copy of the Secure Context ID is incorrectly updated with the value of the Non-Secure Context ID. The Non-Secure Context ID is correctly updated.

If the Secure Context ID is updated in condition 3 above, then when the ETMPWRDOWN bit is cleared the ETM's copy of the Non-Secure Context ID is incorrectly updated with the value of the Secure Context ID. The Secure Context ID is correctly updated.

Implications

The incorrect Context ID is used by the ETM's internal Context ID comparators, resulting in incorrect operation of the ETM when the Context ID comparators are used.

The incorrect Context ID is transmitted in the trace stream when tracing is enabled.

This could result in the Secure Context ID being transmitted.

The incorrect Context ID is used until the affected Context ID is next updated.

Workaround

There is no workaround for this erratum.

398636: AFREADYM signal might not be asserted

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

On the AMBA Trace Port (ATB) of the CoreSight ETM11, two signals are provided to implement a trace data flushing mechanism. AFVALIDM is an input signal which requests a flush of all trace currently stored in the CoreSight ETM11. AFREADYM is an output signal which indicates when all stored data has been output by the CoreSight ETM11 and is asserted in response to an assertion of AFVALIDM. This flushing mechanism allows trace capture devices to dynamically request all stored data to be output.

On CoreSight ETM11 the AFREADYM signal might not be asserted under certain conditions, causing the ETM to never acknowledge the flush request.

Two sets of conditions can cause this erratum to occur, described below in Conditions 1 and Conditions 2.

Conditions 1

The following conditions must occur in the following order for this erratum to occur:

1. The CoreSight ETM11 is reset using the nPORESET input
2. The ETM PWRDOWN bit (bit [0] of the ETM Control Register 0x000) is cleared
3. The ETM PWRDOWN bit is set
4. AFVALIDM is asserted

This erratum does not occur when the CoreSight ETM11 is used in the ETM11CSSingle configuration.

Conditions 2

The following conditions must occur in the following order for this erratum to occur:

1. The ETM PWRDOWN bit (bit [0] of the ETM Control Register 0x000) is cleared
2. The ETM is programmed and tracing is enabled
3. Both the NIDEN and DBGEN input signals are driven LOW while trace data is still in the ETM's FIFO
4. AFVALIDM is asserted

This erratum does not occur when the CoreSight ETM11 is used in the ETM11CSSingle configuration.

Implications

Trace capture devices will usually use the flush mechanism to request all data from the system before stopping trace capture. When this erratum occurs these trace capture devices will never receive a flush completion acknowledgement via AFREADYM and therefore might never stop capturing trace. For example, if the trace capture device is the CoreSight TPIU or CoreSight ETB then trace decompression tools might continue polling the TPIU or ETB waiting for trace capture to stop and it never stops, thereby causing an infinite loop until the device is reset, or a timeout in the tools is triggered.

Workaround

This is a workaround for tools vendors.

If a trace capture device is configured to issue a flush before trace capture stops then:

- If the CoreSight ETM11 is connected to a trace funnel, then the CoreSight ETM11 should be configured to be the lowest priority ATB source. This ensures that the ETM flushed last. This might impact the number of FIFO overflows observed in a bandwidth limited system.
- The tools should monitor the status of the flush operation. When using a CoreSight TPIU or CoreSight ETB this can be done using the Formatter and Flush Status Register. If a flush is observed to continue for a long period of time (for example, 100ms or more) then trace capture should be disabled immediately without waiting for the flush to complete.

This might result in some of the trace which was present in the CoreSight ETM11 to not be captured by the trace capture device.

Trace tools can avoid this erratum by configuring the trace capture devices to never request a flush when the CoreSight ETM11 is connected to the trace capture device. This might result in some of the trace which was present in all trace sources to not be captured by the trace capture device.

398884: Out-of-order data might be incorrect

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

The CoreSight ETM11 might output an incorrect data value in an out-of-order data packet. In addition, the out-of-order data packets corresponding to other out-of-order placeholders might never be output.

There are multiple scenarios in which this erratum can occur, described below in Conditions 1 to Conditions 2.

Conditions 1

1. The ARM11 processor executes one or more data instructions, using all three slots in the ARM11 Load-Store Unit
2. The data for the three slots is returned out-of-order and on three consecutive cycles

When this erratum occurs, the data for the slots which return second and third is corrupted and is traced as a single out-of-order data packet with tag 3 or two out-of-order data packets both with tag 3.

For example, an LDM instruction where the data for the first few transfers is not in the cache and data for later transfers is in the cache might cause this erratum.

Conditions 2

1. The ARM11 processor executes one or more data instructions
2. One of the data values for the first instruction is passed to the ETM before the first instruction is presented to the ETM and this data value is returned out of order

When this erratum occurs the ETM issues an out-of-order placeholder followed by an out-of-order data packet for the same tag on the same cycle, however the data value corresponds to an earlier out-of-order placeholder and trace decompression tools will associate the data value with the wrong address and possibly the wrong instruction. The data value for the second out-of-order placeholder is never returned.

For example, an LDM instruction where the LDM is stalled in the pipeline and where some of the data for the LDM is in the cache and other data is not in the cache might cause this erratum.

Implications

The data value traced is incorrect. Any data value comparisons might behave incorrectly. This erratum does not affect data address tracing. Trace packet synchronisation is maintained.

Workaround

This erratum is not visible if data value trace is not enabled and if no data value comparisons are enabled.

There is no workaround for this erratum.

408879: Periodic Synchronisation might not occur

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p2,r1p0, Fixed in r1p1.

Description

Under normal conditions, the ETM outputs instruction synchronization (I-Sync) and data address synchronization (D-Sync) packets periodically, where the period is defined by the Synchronization Frequency Register (register 0x078). The synchronization packets allow the trace to be decompressed when earlier synchronization packets in the trace are missing, for example when trace is stored in a circular buffer and the oldest trace has been overwritten.

When this erratum occurs, no periodic I-Sync or D-Sync packets occur in the trace stream until the next non-periodic I-Sync packet.

Conditions 1

1. The ETM is reset, usually by a power-on reset of the device
2. The ETM is enabled and the programming bit is cleared in the same cycle as an instruction is executed

It is important to note that this erratum can only occur the first time the programming bit is cleared after a power-on reset.

If the processor is in debug state when the ETM is programmed, this erratum does not occur.

This set of conditions can only cause the erratum on the r0p2 revision of CoreSight ETM11.

Conditions 2

1. The ETM is reset, usually by a power-on reset of the device
2. The ETM Synchronization Frequency register (0x78) has not been programmed
3. The ETM is enabled and the programming bit is cleared in the same cycle as an instruction is executed

It is important to note that this erratum can only occur the first time the programming bit is cleared after a power-on reset.

If the processor is in debug state when the ETM is programmed, this erratum does not occur.

This set of conditions can only cause the erratum on the r0p2 and r1p0 revisions of CoreSight ETM11.

Implications

The absence of periodic I-Sync and D-Sync packets could mean that no I-Sync or D-Sync packets are present in the captured trace. This means that trace decompression is not possible. This will only occur when non-periodic synchronization packets do not occur, for example when continually tracing program execution where no FIFO overflows occur.

If a non-periodic I-Sync packet is generated, then all trace prior to this non-periodic I-Sync might not be decompressible. The non-periodic I-Sync packet would be generated under any of the following conditions:

- Trace is disabled by TraceEnable going LOW
- An ETM FIFO overflow

- The processor halts in debug state
- The ETM is reprogrammed, which involves setting the programming bit

Alignment synchronization (A-Sync) packets are not affected by this erratum.

Workaround

This is a workaround for tools vendors or users.

When clearing the programming bit for the first time, the following sequence is recommended.

1. Program the TraceEnable event register 0x008 with 0x0000406F, which disables trace generation
2. Program the Synchronization Frequency register 0x78 as desired
3. Clear the programming bit in the ETM Control register 0x000
4. Wait for the programming bit to be cleared in the ETM status register 0x004
5. Set the programming bit in the ETM Control register 0x000
6. Wait for the programming bit to be set in the ETM status register 0x004
7. Configure the TraceEnable Event register 0x008 as originally desired
8. Clear the programming bit in the ETM Control register 0x000
9. Wait for the programming bit to be cleared in the ETM status register 0x004

412415: Precisely aborted unaligned transfers might cause unpredictable data address comparator behavior

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

If an unaligned transfer is precisely aborted, address comparators which are configured to match on data addresses might behave Unpredictably. This is because an unaligned transfer is presented to the ETM as two separate but consecutive transfers. If the precise data abort causes only the first of these transfers to be presented then the comparators use the next transfer which is presented to complete the transfer. This next transfer is always related to a future instruction and has no relationship to the original unaligned transfer. This causes the comparator to compare against an incorrect address.

Conditions

The following conditions must occur in order:

1. The ARM11 processor has the U bit set, enabling unaligned transfers
2. A load or store instruction is executed which accesses an unaligned address
3. The first part of the unaligned transfer is presented to the ETM
4. A precise data abort occurs on the unaligned access
5. A subsequent load or store instruction is executed

The first data transfer associated with the load or store instruction in stage 5 is considered by the ETM's comparators to be associated with the unaligned transfer. This data transfer then causes Unpredictable behaviour of any address comparators which are configured for data address comparisons.

Implications

Any ETM resource which is dependent on the comparator might behave Unpredictably on the first data transfer after the precise data abort. This means the comparator might incorrectly match, or incorrectly fail to match.

No subsequent transfers are affected.

This erratum does not result in corrupted trace.

Workaround

There is no workaround for this erratum.

425203: Trigger might not occur

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2.

Description

If an A-Sync packet is generated at the same time as a trigger occurs, the Trigger packet might not appear in the trace stream.

Additionally, the TRIGOUT signal from the ETM11CS might not be asserted.

Conditions

The following conditions must occur in the same clock cycle:

1. An A-Sync packet is generated
2. A Trigger occurs

A-Sync packets are generated periodically in the trace stream and the period is dependent on the value of the Synchronization Frequency Register (register 0x078), which can take a value of between 60 and 4096.

Implications

The Trigger packet does not occur in the trace stream.

If using an ETM11CSSingle, the trigger condition might not occur on the trace port. This means the trace capture and analysis tools might not stop capturing trace which means that the desired trace might be overwritten in the trace capture device.

If using the CoreSight ETM11 in a CoreSight system the TRIGOUT signal is not asserted. This means that the trigger condition might not occur on the trace port or be embedded in the trace stream by a Trace Port Interface Unit. This means trace capture and analysis tools might not stop capturing trace which means that the desired trace might be overwritten in the trace capture device. Additionally, any cross triggering in the system which is configured to trigger on the ETM's TRIGOUT signal might not trigger.

It should be noted that this erratum is unlikely to occur. This is because A-Sync packets are usually generated quite rarely, depending on the Synchronization Frequency. For example, the default value of the Synchronization Frequency Register is 1024, which means that an A-Sync packet is generated every 1024 bytes of trace. Since the ETM can generate up to 4 bytes of trace per cycle, this would imply a 1 in 256 chance of the erratum occurring in a single trace run. However, since the trace output is bursty, with an average of around 2 bytes per cycle for full data trace, the erratum is likely to occur significantly less than once in 256 separate trace runs with the default Synchronization Frequency Register value (1024).

Workaround

The probability of this erratum occurring can be reduced by increasing the value of the Synchronization Frequency Register.

There is no workaround for this erratum when using ETM11CSSingle.

If using CoreSight ETM11 in a CoreSight system, then an External Output (EXTOUT) from the ETM could be used to trigger the cross-trigger infrastructure and the trace sink, such as a TPIU or ETB. It should be noted that the normal trigger generation in the ETM only generates one trigger event, whereas using the EXTOUT might

cause multiple trigger events if used in this way. The ETM's sequencer could be used in conjunction with the EXTOUT to avoid this.

426912: Incorrect byte lanes selected in Big-Endian mode

Status

Affects: product CoreSight ETM11.

Fault status: Cat 2, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

The ETM address comparators can match data addresses against a pre-programmed value. The Address Comparator Access Type registers control the type of access which should be compared against, allowing the address comparators to match only when a particular type or size of access is performed to the programmed address location.

When the processor is operating in BE-32 big-endian mode, the address comparators might behave incorrectly when a half-word or byte access is performed by the processor. This is because the comparators interpret the memory system's view of the address, when they should be interpreting the processor's view.

Conditions for Single Address Comparators

1. The processor is operating in BE-32 big-endian mode
2. The comparator is programmed to watch a byte or a half-word (bits [4:3] of the Access Type Register, 0x020-0x02F)
3. The processor performs a half-word or byte data transfer to an address which is in the same word as the watched address

For example, consider when the comparator is programmed to watch a byte at address 0x1000. If a byte is accessed at 0x1000, then the byte at 0x1003 is accessed in memory and the address comparator will incorrectly fail to match. If a byte is accessed at 0x1003, then the byte at 0x1000 is accessed in memory and the address comparator will match incorrectly.

This erratum does not affect the CoreSight ETM11 when connected to the ARM11 MPCore because the ARM11 MPCore does not support BE-32 mode.

Conditions for Address Range Comparators

1. The processor is operating in BE-32 big-endian mode
2. One of the addresses of an address range comparator is not aligned to a word address
3. The processor performs a half-word or byte data transfer to an address which is in the same word as the non-word-aligned address programmed into the comparator

For example, consider when the comparator is programmed with a low address of 0x1000 and a high address of 0x100E. If a byte is accessed at 0x100C, then the byte at 0x100F is accessed in memory and the address comparator will incorrectly fail to match. If a byte is accessed at 0x100F, then the byte at 0x100C is accessed in memory and the address comparator will match incorrectly.

If the Address Range comparator addresses are word-aligned, this erratum does not affect Address Range comparators.

This erratum does not affect the CoreSight ETM11 when connected to the ARM11 MPCore because the ARM11 MPCore does not support BE-32 mode.

Implications

This erratum only occurs when the processor is configured in BE-32 big-endian mode.

The Address Comparator might match unexpectedly or might unexpectedly fail to match.

Any events which have been programmed to be sensitive to the Address Comparator might behave unexpectedly. This might include a trigger occurring incorrectly, or trace might be missing or generated unexpectedly.

Workaround

This is a workaround for tools vendors and users.

If the processor is known to always operate in BE-32 mode permanently, then Single Address Comparators for half-word or byte accesses must be programmed with the memory view of the address:

- If the comparator is programmed for a byte access, bits [1:0] of the programmed address must be inverted.
- If the comparator is programmed for a half-word access, bit [1] of the programmed address must be inverted

If the processor dynamically switches between BE-32 and other endian modes, there is no workaround for this erratum for Single Address Comparators.

Address range comparators should be programmed with word-aligned addresses

Errata - Category 3

345320: Branch Output bit can be set, but feature not supported

Status

Affects: product CoreSight ETM11.

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

Description

ETM control register (0x00) bit [8] is the Branch Output bit. This bit controls a feature not present in CoreSight ETM11, so should always read as zero. However, this bit has been erroneously left as modifiable. Tools may therefore incorrectly infer that the Branch Output feature is present.

Conditions

Testing for the presence of the Branch Output feature of the ETM by checking that bit [8] of the ETM control register is modifiable.

Implications

Tools may incorrectly infer that the Branch Output feature is present in CoreSight ETM11 r0p0.

Workaround

The work-around is for tool vendors.

Testing for the presence of the Branch Output feature in CoreSight ETM11 r0p0 is not possible via the Branch Output bit. Checking the ETM ID register value is an alternative method for identifying whether this feature is present.

Implications of workaround

The code size is increased.

345449: Unaligned big-endian (BE-32) data transfer addresses inferred incorrectly by ETM

Status

Affects: product CoreSight ETM11.

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

Description

When in BE-32 big-endian mode, address bits [1:0] of unaligned data access are reported incorrectly by the ETM. Note that unaligned BE-32 data access is UNPREDICTABLE in the ARMv6 architecture, so should be avoided.

Conditions

The following conditions must be met:

1. The core is configured for word-invariant big-endian data accesses
2. Unaligned data accesses are enabled
3. The address of an unaligned data access is traced

Implications

Bits [1:0] of the data address of BE-32 unaligned data accesses are traced incorrectly. Note that this type of access is Unpredictable in ARMv6.

Workaround

Avoid unaligned accesses in the word-invariant BE-32 configuration as these are architecturally Unpredictable. To do this enable strict alignment checking by setting the A bit of CP15 c1 to 1.

345456: Data missed from trace when next to branch-phantom at start of include region with exact-match bit set in data-only mode

Status

Affects: product CoreSight ETM11.

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

Description

A branch phantom is executed with an address equal to the address in a comparator programmed as an include region for TraceEnable. The access type registers are programmed for instruction execute with the exact-match bit set, in data-only mode. On the same cycle, a data instruction is executed whose address is also in the correct range. The ETM fails to trace the data even though it should.

Conditions

The following conditions must be met:

1. A branch phantom is executed with address equal to address comparator
2. On the same cycle, a data instruction is executed
3. Access type set for instruction execute with exact match bit set
4. Tracing in data-only mode

Implications

A data item may be missing from trace.

Workaround

Avoid setting the exact match bit for address comparators used for include regions in data-only mode.

Implications of work-around

Without the exact match bit set, cancelled instructions are included in the traced region.

However cancelled instructions should have no associated data, so this should have no impact in data-only mode.

348902: Interrupted stores might be unexpectedly traced

Status

Affects: product CoreSight ETM11.

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

Description

If a data instruction is interrupted by an FIQ or IRQ it might be cancelled, depending on the point at which the interrupt occurs. If it is cancelled all data items which it would have transferred must be discarded because the instruction must be treated as if it never occurred. If the cancelled instruction would normally have been followed by a store instruction, then transfers for the store instruction (which is already in the ARM11 pipeline) might be traced unexpectedly.

Conditions

This erratum can only occur when the ARM11 CPU is operating in Low-Interrupt-Latency mode.

For example:

Normal uninterrupted execution:

Actual and Traced Execution
0x1000: Instruction: LDR r0, [r1]
Load from 0x2000
0x1004: Instruction: STMIA r4, {r8-r10}
Store to 0x3000 (r8)
Store to 0x3004 (r9)
Store to 0x3008 (r10)
0x1008: Instruction

Interrupted execution:

Actual Execution	Traced Execution	Comments
0x1000: Instruction: LDR r1, [r0]	0x1000: Instruction: LDR r1, [r0]	
Out-of-Order Placeholder for transfer from 0x2000	Out-of-Order Placeholder for transfer from 0x2000	
	Store to 0x3000 (r8)	This is incorrectly traced
FIQ occurred cancelling the instruction at 0x1000 and preventing the return data	FIQ occurred cancelling the instruction at 0x1000 and preventing the return data	This is correctly traced

transfer from 0x2000	transfer from 0x2000	
0x18: Instruction: FIQ exception vector (LDR PC, [0x58])	0x18: Instruction: FIQ exception vector (LDR PC, [0x58])	This is correctly traced

Implications

Instruction trace is unaffected.

An extra data item might be traced for a data instruction before an interrupt which cancelled the instruction.

Resources which depend on data transfers might behave incorrectly.

Workaround

This workaround is for tool vendors.

If an instruction which causes a single data transfer (for example an LDR instruction) has multiple data transfers associated with it when the trace is decompressed, any additional transfers must be discarded. This workaround only needs to be applied when a cancelling exception is received, which cancels the aforementioned instruction.

Implications of workaround

Tools must keep a history of executed data instructions and data transfers to be able to discard these extra data transfers.

349882: Writes to counter value register might cause counter event to fire immediately

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1, Fixed in r0p2.

Description

If the counter reload value register (register 0x50 or 0x51) is written with a value of zero and the counter value register (register 0x5c or 0x5d) is written with a value greater than zero, the counter should decrement on the counter enable event and then fire the 'counter at zero' event once the counter reaches zero. This erratum causes the 'counter at zero' event to fire immediately after the programming bit is cleared.

Conditions

1. Counter reload value register (0x50 or 0x51) programmed with zero
2. Counter value register (0x5c or 0x5d) programmed with a non-zero value

Implications

The counter behaves unexpectedly.

Workaround

There is no workaround for this erratum and tools vendors are advised to avoid programming the counters in this manner.

This is outside the normal usage model of the counter. The counter value register (0x5c or 0x5d) should only be written to restore the value of the counter if the state of the ETM is saved to memory and restored later. Under these conditions the counter value register is always less than or equal to the value of the counter reload value register and the erratum does not occur.

364310: Authentication Status register value is incorrect

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1, Fixed in r0p2.

Description

The value of the Authentication Status register (register 0x3EE, offset 0xFB8) is incorrect when the CoreSight ETM11 is connected to an ARM1176JZ-S or ARM1176JZF-S processor. This read-only register indicates the level of tracing permitted.

Bits [7:6] of the register take the value 2'b10 or 2'b11, but should take the value 2'b00.

Bits [3:2] of the register take the value 2'b00, but should take the value 2'b10 or 2'b11, as seen in bits [7:6].

The register should read as follows:

Bits[31:4] are LOW

Bit[3] is HIGH

Bit[2] is the logical OR of the DBGEN and NIDEN input signals to the CoreSight ETM11

Bits[1:0] are LOW

Implications

Trace analysis tools might incorrectly interpret the level of trace supported by the ETM.

Workaround

This is a workaround for tools vendors.

If the bit [19] of the ETM ID register (0x079) is HIGH, this indicates that the CoreSight ETM11 is connected to the ARM1176JZ-S or ARM1176JZF-S and this workaround must be applied.

When reading the Authentication Status register (0x3EE), the value of bits [7:6] should be interpreted as if they were present in bits [3:2] of this register. Bits [7:6] of the register should be LOW, indicating that secure non-invasive debug is not controlled by the ETM.

374728: Lock Access can be modified when PADDRDBG31 is HIGH

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When PADDRDBG31 is HIGH, the CoreSight ETM11 should consider all programming accesses to have been initiated by an external debugger. When PADDRDBG31 is LOW, the CoreSight ETM11 should consider all programming accesses to have been initiated by software running on the system.

The lock access mechanism consists of the Lock Access Register (register 0x3EC) and the Lock Status Register (register 0x3ED). The Lock Status Register should indicate that no lock access mechanism exists if programming accesses are initiated from an external debugger, i.e. when PADDRDBG31 is HIGH. Writes to the Lock Access register should be ignored by the ETM when PADDRDBG31 is HIGH.

Due to this erratum on CoreSight ETM11, when PADDRDBG31 is HIGH, the Lock Status Register correctly indicates that no lock access mechanism is present on accesses from an external debugger. However the Lock Access Register can be used to mistakenly allow accesses by software running on the system. Also, the Lock Access Register can be used to mistakenly prevent accesses by software running on the system.

Conditions

1. PADDRDBG31 is HIGH, indicating an access from an external debugger
2. Debugger software writes to the Lock Access Register without checking the Lock Status Register

Implications

It is possible for target resident software to be given access to the ETM registers when accesses should be prevented using the lock access mechanism. Conversely, it is possible for target resident software to be prevented from accessing the ETM registers when accesses should be allowed by the lock access mechanism.

If an external debugger writes the lock access key, 0xC5ACCE55, to the Lock Access Register, the software lock is unlocked and software running on the system can write to the ETM registers. Additionally, if an external debugger was to write any value apart from 0xC5ACCE55 to the Lock Access Register, the software lock is locked, preventing any software running on the system from writing to the ETM registers.

Since an external debugger should read the Lock Status Register to determine if the lock mechanism is present before writing to the Lock Access Register, it is expected that no external debugger will write to the Lock Access Register thereby never causing this erratum to have any adverse effects.

Workaround

This is a workaround for tools vendors.

Tools must follow the recommended behavior for operating the lock access mechanism. Tools must read the Lock Status register before writing to the Lock Access Register. If the Lock Status Register indicates that no lock access mechanism exists, then the debugger must not write to the Lock Access Register.

This is the normal recommended behavior for tools, therefore in most cases no workaround is necessary.

398993: Cycle Accurate Count inaccurate on exit from overflow

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1, Fixed in r0p2.

Description

When the CoreSight ETM11 FIFO overflows, tracing stops and the FIFO is emptied. Once the FIFO is empty, the ETM restarts tracing and generates an I-Sync packet indicating the current program location. When cycle accurate tracing is enabled, the I-Sync packet contains a cycle count indicating the number of cycles the ETM was in overflow. When this erratum occurs the cycle count might be one cycle greater than the real number of cycles spent in overflow.

Conditions

The following conditions must all be true on the same cycle:

1. The CoreSight ETM11 is enabled and cycle accurate tracing is enabled
2. Two instructions are executed where the first instruction passed its condition code check and the second instruction failed its condition code check
3. The ETM FIFO overflows

Two instructions might be executed in a single cycle under the following conditions:

- a branch is folded onto another instruction
- an IT instruction is executed (ARM1156TS(F)-S only) and folded onto the next instruction
- some Thumb 16-bit NOP instructions are executed and folded onto the next instruction

Implications

The cycle accurate count is greater than the real number of cycles by 1 cycle. Tools using the cycle accurate count to profile code will be slightly inaccurate around ETM FIFO overflows. It is expected that overflows should be avoided and therefore this erratum is likely to occur rarely.

Workaround

There is no workaround for this erratum.

400435: ViewData programmed to be sensitive to a non-data instruction might behave incorrectly

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1, Fixed in r0p2.

Description

Under normal conditions, when ViewData is set to be sensitive to instruction address comparisons one of the following should happen:

1. If ViewData is set to exclude an instruction then all data for that instruction should not be traced
2. If ViewData is set to include an instruction then all data for that instruction should be traced, assuming no other exclude condition exists

This erratum might cause data to unexpectedly be included or excluded if the data occurs in the same cycle as a non-data instruction.

Conditions 1 - Exclude

1. One or more address range comparators are set to match on a range of instruction addresses
2. ViewData is set to exclude one or more of the address range comparators
3. The address range comparator matches on an instruction which is not a data instruction
4. Data for an earlier instruction which was included for ViewData occurs in the same cycle as the non-data instruction

Under these conditions, if the earlier instruction was supposed to be included by ViewData, then all data for that instruction should be included. When these conditions occur, the data in stage 4 above is not traced.

Conditions 2 - Include

1. One or more address range comparators are set to match on a range of instruction addresses
2. ViewData is set to include one or more of the address range comparators
3. The address range comparator matches on an instruction which is not a data instruction
4. Data for an earlier instruction which was excluded from ViewData occurs in the same cycle as the non-data instruction

Under these conditions, if the earlier instruction was supposed to be excluded by ViewData, then all data for that instruction should be excluded. When these conditions occur, the data in stage 4 above is traced unexpectedly.

Implications

Data trace is unexpectedly missing or is unexpectedly traced.

This erratum is unlikely to occur very often because data rarely returns very late, and the probability of it also returning when moving from a ViewData include region to a ViewData exclude region (or vice versa) is very small.

Workaround

There is no workaround for this erratum.

404712: Incorrect Context ID and security state might be reported at trace turn-on

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2,r1p0,r1p1, Open.

Description

Under normal conditions, when trace turns on an I-Sync packet is generated which includes:

- Instruction address
- Processor state
- Security state
- Context-ID (optional)
- Reason for trace turn-on

If trace turns on while an earlier data transfer instruction is still executing, then the address in the I-Sync packet is that of the outstanding data instruction, since some data for that instruction might be returned and traced.

For this erratum to occur, trace must turn on while an earlier data instruction is still executing and just after the security state has changed. When the erratum occurs the security state and Context-ID of the data instruction are incorrectly reported and are from the wrong security state.

Conditions

The following conditions must occur in order:

1. The CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor
2. The CoreSight ETM11 is enabled but is not actively tracing
3. The ARM11 processor executes a data transfer instruction, but the data takes a long time to return (e.g. cache miss)
4. The ARM11 processor changes security state and executes at least one instruction in the new security state
5. The ETM11 starts tracing
6. The data for the instruction in Condition 3 is returned or a new data transfer instruction is executed

It is important that no data instructions are executed between Conditions 3 and 6.

Implications

When this erratum occurs, the I-Sync packet generated at the start of trace indicates the wrong security state and Context-ID for the outstanding data instruction.

For example, if the security state change was from Secure to Non-Secure, the I-Sync packet generated at Condition 5 above incorrectly reports the data instruction as executing in Non-Secure state and the Context-ID traced is the Non-Secure Context-ID. The data instruction actually executed in Secure state.

If the security state change was from Non-Secure to Secure, the I-Sync packet generated at Condition 5 above incorrectly reports the data instruction as executing in Secure state and the Context-ID traced is the Secure Context-ID. The data instruction actually executed in Non-Secure state.

All subsequent instructions traced are reported in the correct security state with the correct Context-ID, since all subsequent instructions are executed in the new security state.

This erratum does not have any security implications, since if the data instruction executed in a region where tracing was prohibited then it would not be traced when tracing was enabled in Condition 5 above.

This erratum is very unlikely to occur because of all the conditions which would have to be met. Typically, turning trace on part-way through a data instruction is unlikely, combining that with a security state change which occurs while the data instruction is still outstanding makes the conditions very unlikely to occur.

Workaround

There is no workaround for this erratum.

407022: Extra Store-Failed packet produced when using ViewData to filter store-exclusive data transfers

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p2, Fixed in r1p0.

Description

When tracing a failed store-exclusive data transfer, a data packet is generated by the CoreSight ETM11, followed by a Store-Failed packet. For STREXD instructions which fail, 2 data packets are generated and each one is followed by a Store-Failed packet.

Using the ViewData resource in the CoreSight ETM11 allows zero, one or both data packets of the failed STREXD data transfers to be traced. If ViewData is set to only trace the second data transfer, this erratum causes two Store-Failed packets to be generated after the data packet is traced instead of one.

Conditions

1. The CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor or an ARM1136J(F)-S r1 processor.
2. The CoreSight ETM11 is configured to trace data addresses, data values or both.
3. ViewData is configured to use one or more address comparators as include or exclude regions. This might mean that single addresses are excluded or included, or ranges of addresses are included or excluded.
4. The ARM11 processor executes a failed STREXD instruction. The above ViewData configuration causes the upper word access to be traced and the lower word access not to be traced.

For example, ViewData might be programmed to include Single Address Comparator 1. Single Address Comparator 1 is programmed to match when a data transfer to address 0x1004 is performed. If a failed STREXD instruction accesses the 32-bit words at 0x1000 and 0x1004 then the ViewData configuration would cause only the second transfer at 0x1004 to be traced, thereby causing this erratum.

Implications

If this erratum occurs, two Store-Failed packets are produced after the data packet. The second packet is redundant because the failed transfer has already been indicated.

Workaround

This is a workaround for tools vendors.

If two Store-Failed packets are encountered in consecutive packets, the second Store-Failed packet must be ignored.

409113: ETMJTAGPORT behaves incorrectly when simultaneous JTAG and APB transactions occur

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

The ETMJTAGPORT, as used in ETM11CSSingle, supports accesses originating from a JTAG port or from a system-driven APB port. The system-driven APB port will typically be driven by target resident software. In most situations, only one of these access mechanisms will be used at any one time.

For this erratum to occur, an access from the system-driven APB port must occur at the same time as the JTAG port is in use. When this erratum occurs, Unpredictable behavior might occur on the APB input port to the CoreSight ETM11.

Conditions 1

The following conditions must occur in order:

1. Scan chain 6 is selected in the ETMJTAGPORT
2. INTEST is selected as the JTAG instruction in the ETMJTAGPORT
3. A software access is initiated over the system-driven APB input port to the ETMJTAGPORT
4. Scan chain 6 is deselected, or a JTAG instruction which is not INTEST is selected in the ETMJTAGPORT

The software access in stage 3 above is delayed because the PREADY output from the ETMJTAGPORT is driven LOW. After stage 4 above, the ETMJTAGPORT allows the software access to be transmitted to the CoreSight ETM11 and subsequently allows the software access to complete.

This erratum can only occur if the ETM11CSSingle configuration is used.

Conditions 2

The following conditions must occur in order:

1. Scan chain 6 is selected in the ETMJTAGPORT
2. A software access is initiated over the system-driven APB input port to the ETMJTAGPORT
3. PCLK cycle after the software access is initiated, INTEST is selected as the JTAG instruction in the ETMJTAGPORT

The software access in stage 2 above is delayed because the PREADY output from the ETMJTAGPORT is driven LOW.

This erratum can only occur if the ETM11CSSingle configuration is used.

Implications

System-driven accesses to the ETM using the APB port might cause Unpredictable behavior if performed during a debugger-driven JTAG access. This behavior might include:

- The software access in Conditions 1 above completes incorrectly. If the transaction was a write, the register might not be updated.
- If the transaction was a read, the read data returned might be incorrect.
- The APB protocol on the input to the CoreSight ETM11 might be violated, causing subsequent accesses to the ETM to behave incorrectly. This might involve PREADYDBG from the CoreSight ETM11 being held LOW, causing a deadlock on the APB interface, preventing any further APB operations to the ETM or any other components on same the APB bus.
- Future JTAG driven accesses might have no effect, in that write accesses do not update the ETM and read accesses return Unpredictable values.

In normal circumstances, only one access mechanism is used at any one time. Therefore this erratum is very unlikely to occur.

Workaround

This is a workaround for tools vendors.

To prevent this errata occurring, only one of the access mechanisms must be used at any one time, either JTAG or software accesses. If this is the case, this erratum does not occur.

410117: Address Range comparator matches incorrectly on 64-bit transfer

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

The CoreSight ETM11 treats 64-bit data transfers as two separate 32-bit data transfers. When the internal address comparators compare a data transfer address, a result is produced for each of the 32-bit transfers.

If an address range comparator is set to match on data addresses where the upper address is on a non-word-aligned address, the second 32-bit transfer might match incorrectly.

Conditions

1. An address range comparator is programmed with bit [2] of the upper address comparator set to b0 and bits [1:0] are not equal to b00
2. A 64-bit transfer is performed where the address bits [31:3] match the programmed address bits [31:3]
3. The address range comparator is configured to be used by ViewData

For example, if the address range comparator is programmed with a low address of 0xE00 and an upper address of 0xF01, only transfers which access any bytes from 0xE00 to 0xF00 inclusive will cause the comparator to match. If a 64-bit transfer occurs to address 0xF00, the bytes accessed are as follows:

1. 1st 32-bit transfer accesses 0xF00 to 0xF03. This correctly results in a match because address 0xF00 is accessed.
2. 2nd 32-bit transfer accesses 0xF04 to 0xF07. This incorrectly results in a match.

Implications

This causes the address range comparator to incorrectly match on the 2nd 32-bit transfer. The address range comparator always correctly matches on the 1st 32-bit transfer, because this is in the range.

The only visible effect this has on ETM behavior is when the address range is configured to be used by ViewData.

If the address range comparator is excluded from ViewData, the 2nd 32-bit transfer is incorrectly excluded, even though it was outside the range.

If the address range comparator is included by ViewData, the 2nd 32-bit transfer is incorrectly included, even though it was outside the range.

It is expected that condition 1 above is rarely programmed, since ranges are typically programmed to operate on word-aligned boundaries. If ranges are always programmed with word-aligned addresses, this erratum does not occur.

Workaround

There is no workaround for this erratum.

To avoid unexpected behaviour of the address range comparators, ensure that word-aligned addresses are always used.

411183: Address Range comparator with Exact Match bit set fails to match after a coprocessor access

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

Under normal conditions, when an address comparator is programmed with the exact match bit set (bit [7] of the Address Access Type register), if an out-of-order data transfer occurs the comparator waits until the data value is returned by the processor before matching. Setting the exact match bit is useful when you want the comparator to match only when it is sure that the transfer does not data abort or store-fail, or if you are performing data value comparisons.

For this erratum to occur, a coprocessor access must occur between the out-of-order placeholder and the out-of-order data. If this occurs, the address comparator will not match when the out-of-order data returns, even if the address matched.

Conditions

The following conditions must occur in order:

1. A Single Address comparator or an Address Range comparator is programmed with the exact match bit set to b1
2. An out-of-order placeholder occurs which matches the Single Address comparator or Address Range comparator. The address comparator does not fire immediately because it must wait for the out-of-order data to return.
3. A coprocessor access is performed.
4. The out-of-order data for the earlier placeholder returns.

The address comparator does not fire in condition 4 above.

Implications

Any resources which are programmed to be sensitive to an address comparator with the exact match bit set might behave unexpectedly.

Typically, when the exact match bit is set, the comparator is only used to control resources such as the trigger, counters, sequencer or external outputs. For example, a trigger might not occur, a counter might not decrement, the sequencer might not change state or an external output might not be correctly asserted.

If the exact match bit in the comparator is clear, the comparator will correctly always fire in condition 1 above, however if the transfer data aborted or store-failed or if a data value comparison was being performed this might not be the desired operation. This might lead to unwanted triggers, counter miscounts, erroneous sequencer state changes or unexpected assertion of external outputs.

Workaround

There is no workaround for this erratum.

411191: TraceEnable Start/Stop resource stops early

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p2, Fixed in r1p0.

Description

Under normal conditions, the TraceEnable Start/Stop block is calculated for every item on a cycle in the order in which they occur. For example, all instruction items occur before any data items in a single cycle.

For this erratum to occur, the start/stop resource must be active and a stop address is encountered on an item which is not the first item in a cycle. The start/stop event resource incorrectly stops in this cycle.

Conditions

The following conditions must occur in order:

1. A start address is encountered
2. A stop address is encountered in a cycle where the matching item is not the first item in the cycle.

For example, consider when an instruction and a data transfer occur in the same cycle, where the data transfer matches a stop address. The instruction occurs before the data transfer, so the start/stop resource must be active on the instruction. This erratum causes the start/stop event resource to stop on this cycle, where it should stop on the following cycle.

Implications

This erratum causes any events which are sensitive to the start/stop event resource to behave unexpectedly. The impact of this could be a counter counting for too few or too many cycles. Since the start/stop resource is usually active for multiple cycles under most conditions, this is the most likely impact of this erratum.

TraceEnable is unaffected by this erratum, unless the start/stop resource is selected through the TraceEnable event register (register 0x008).

Workaround

There is no workaround for this erratum.

411334: Address Range comparator is not cleared by a coprocessor access

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

Under normal conditions, Address Range comparators configured for data address comparisons hold their state between accesses. If a coprocessor access occurs, the comparator should stop firing because there is no address to compare against.

This erratum causes the ETM to ignore the coprocessor access and the address range comparator will continue to fire.

Conditions

The following sequence must occur in order:

1. A data address range comparator is configured with the exact match bit clear
2. A data transfer occurs inside the range, causing the comparator to fire
3. An out-of-order coprocessor access occurs

An out-of-order coprocessor access is usually only possible when certain coprocessor data transfer instructions are followed by another data instruction or another coprocessor data transfer instruction.

The out-of-order coprocessor access should clear the Address Range comparator, however this erratum causes the Address Range comparator to continue firing.

Implications

Any ETM resource which is configured to be sensitive to the comparator might behave unexpectedly, in that a counter might count for longer than expected or tracing might be disabled or enabled for longer than expected.

Trace is not corrupted by this erratum.

Workaround

There is no workaround for this erratum.

412312: Data address comparator might match incorrectly around folded branch instruction

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

This erratum only occurs when the exact match bit (bit [7] of the Address Access Type Register) is set to b1. When a single address comparator is set to match on a data address with the exact match bit set, if a folded branch occurs near an access to the comparison address the comparator might match unexpectedly.

Three sets of conditions might cause this erratum to occur, as described in Conditions 1, Conditions 2 and Conditions 3.

Conditions 1

The following sequence must occur in order:

1. A data address comparator is configured with the exact match bit set to b1
2. A branch instruction is folded onto a load or store instruction
3. The load or store instruction accesses the address specified in the address comparator
4. The data access causes an out-of-order placeholder in the same cycle as the instruction

The single address comparator will incorrectly match in stage 4, whereas it should wait for the out-of-order data to return before matching. The comparator will also match when the out-of-order data returns.

There are multiple ways to cause an out-of-order placeholder, including but not limited to:

- There is another data instruction in the following cycle and the data access in stage 3 is not available in the cache
- The data instruction is an LDM or STM which transfers more than 5 registers

Conditions 2

The following sequence must occur in order:

1. A data address comparator is configured with the exact match bit set to b1
2. A data instruction is executed which accesses the address specified in the address comparator
3. The data access causes an out-of-order placeholder
4. A branch instruction is folded onto another instruction
5. The out-of-order data value is returned

The single address comparator will incorrectly match in stage 4, whereas it should wait for the out-of-order data to return in stage 5 before matching. The comparator will also match when the out-of-order data returns.

There are multiple ways to cause an out-of-order placeholder, including but not limited to:

- There is another data instruction in the following cycle and the data access in stage 2 is not available in the cache

- The data instruction is an LDM or STM which transfers more than 5 registers

Conditions 3

The following sequence must occur in order:

1. A data address comparator is configured with the exact match bit set to b1
2. The comparator is set to match or not match the data value (bits [6:5] of the Address Access Type register are not equal to b00)
3. A branch instruction is folded onto a load or store instruction
4. The load or store instruction accesses the address specified in the address comparator, in the same cycle as the load or store instruction.

If the data value comparison was set to match the data value, the comparator will incorrectly fire if the data value in stage 4 does not match the comparison.

If the data value comparison was set not to match the data value, the comparator will incorrectly fire if the data value in stage 4 matches the comparison.

Implications

In Conditions 1 and 2, the single address comparator fires earlier than expected.

In Conditions 3, the single address comparator fires incorrectly.

This means that any resources which use the single address comparator might behave unexpectedly.

This erratum does not affect address range comparators.

Workaround

This is a workaround for users or for tools vendors.

When programming a data address comparator to match a single address with the exact match bit set, use an address range comparator.

412335: ViewData might behave unpredictably if sensitive to a Single Address Comparator with the Exact Match bit set

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

According to the ETM architecture, imprecise tracing might occur if ViewData is set to be sensitive to an address comparator with the exact match bit set. This means that a desired data transfer might not be traced, or an unwanted transfer might be traced.

This erratum causes unrelated transfers to be unexpectedly traced or missed if the exact match bit (bit [7] of the Address Access Type register) is set.

Conditions

The following sequence must occur in order:

1. A data address comparator is configured with the exact match bit set to b1
2. A load or store instruction is executed, which causes an out-of-order placeholder to be generated
3. In the same cycle as stage 2, out-of-order data for an earlier transfer is returned
4. The out-of-order data causes the comparator to match

Implications

If ViewData is configured to exclude the comparator, then the out-of-order placeholder in stage 2 is incorrectly not traced.

If ViewData is configured to include the comparator, then the out-of-order placeholder in stage 2 is incorrectly traced, unless another exclude condition applies.

This erratum does not affect address range comparators.

Workaround

This is a workaround for users or for tools vendors.

When programming a data address comparator to match a single address with the exact match bit set, use an address range comparator.

413518: Address Range comparator with exact match bit set might not match around unaligned transfers

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

Under normal conditions, an unaligned transfer is presented to the ETM as two separate but consecutive transfers. If the out-of-order data for an earlier transfer returns between these two parts of an unaligned transfer, address range comparators might not match.

Conditions

The following conditions must occur in order:

1. The ARM11 processor has the U bit set, enabling unaligned transfers
2. An address range comparator is configured for data address comparisons with the Exact Match bit set
3. A load or store instruction is executed where the data transfer is inside the specified address range
4. A load or store instruction is executed which accesses an unaligned address, where the data address is greater than the upper address of the address range comparator
5. The first part of the unaligned transfer is presented to the ETM
6. The out-of-order data for the instruction in stage 3 is returned
7. One of the following occurs:
    a. The second part of the unaligned transfer is presented to the ETM
    b. An exception occurs which causes all transfers to be discarded
    c. A precise data abort occurs

When the data returns in stage 6, the address range comparator should match because the accessed address was inside the specified range. This erratum causes the range comparator not to match.

Implications

Any ETM resource which is dependent on the comparator might behave unexpectedly on the data transfer in stage 6 because the comparator fails to match. No subsequent transfers are affected.

This erratum does not result in corrupted trace.

If the exact match bit is not set, this erratum does not occur.

The exact conditions for this erratum to occur are unlikely to occur very often because of the precise timing required for the out-of-order data in stage 6. This will only typically occur when the unaligned transfer is precisely aborted, for example when strict alignment checking is enabled in the ARM11 processor.

Workaround

There is no workaround for this erratum.

To avoid this erratum, the exact match bit should be clear, however this might result in unwanted matches of the address range comparator which are normally masked by setting the exact match bit.

413723: PCLKENDBG not used in claim tag register

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Fixed in r1p0.

Description

The clock enable PCLKENDBG for the APB interface is not used on writes to the Claim Tag registers. This causes multiple writes to the Claim Tag registers.

Conditions

1. PCLKENDBG is used to slow down PCLKDBG
2. A write to the Claim Tag Set or Claim Tag Clear register is performed

Implications

This does not cause any functional problems with the operation of the ETM. The write data bus PWDATADBG must be stable for the whole duration of the APB transaction, which means that the same value is written multiple times.

This erratum might cause static timing analysis failures during implementation.

If ETM11CSSingle is implemented and the system-driven APB interface is not used, this erratum cannot occur.

Workaround

This is a workaround for system implementors.

If a slow APB clock is required, PCLKDBG must be gated externally to the ETM to provide the required clock frequency. PCLKENDBG must be tied HIGH.

416413: Data value trace might be missing around unaligned stores

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r1p0, Fixed in r1p1.

Description

If an unaligned store instruction is followed by another data transfer instruction, the data value for the second data instruction might not be traced.

This erratum only occurs when the CoreSight ETM11 is used with the ARM11 MPCore.

Conditions

The following conditions must occur in order:

1. The U bit is set in the System Control Coprocessor, enabling unaligned transfers
2. A data instruction is executed which stores to an unaligned address
3. A second data instruction is executed, where the data transfer uses the same Load/Store Unit slot as the unaligned transfer
4. The data for the second data instruction is returned one cycle after the data for the unaligned transfer

An out-of-order placeholder is traced for the second data instruction, including the address if data address tracing is enabled. The data value for the second data instruction is not traced.

Implications

The data value for the instruction in stage 2 is never traced.

Data value comparators will never match on the data for the instruction in stage 2. Any resources which are dependent on a data value comparator match might not behave as expected.

This erratum does not result in corrupted trace.

Workaround

There is no workaround for this erratum.

424963: ETMJTAGPORT prevents System APB accesses when nTRST is low

Status

Affects: product CoreSight ETM11.

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

Description

The ETMJTAGPORT inside ETM11CSSingle arbitrates between system (APB) programming accesses to the ETM and debugger (JTAG) programming accesses to the ETM. If the JTAG interface is held in reset with nTRST low, system accesses are stalled using the PREADY signal.

Conditions

This erratum occurs in the ETM11CSSingle macrocell only.

1. The nTRST signal on the JTAG interface is held low
2. An access using the system APB interface is attempted

This erratum will not occur in a system which does not support software access to the ETM registers

Implications

The system APB interface is stalled until nTRST is driven high. This might cause any software running on the system to lock up because the bus might be indefinitely stalled.

It should be noted that nTRST should only be asserted for short periods to reset the JTAG interface, thereby only causing temporary stalls of the system APB while the JTAG interface is being reset. nTRST should be held high in idle conditions.

Workaround

There is no workaround for this erratum

425205: Multiple trigger requests might occur

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1, Fixed in r0p2.

Description

The TRIGOUT signal from the ETM11CS might be asserted multiple times for one trigger event.

Conditions

This erratum does not happen in an ETM11CSSingle system.

The following conditions must occur:

1. The TRIGSBYPASS input signal is tied LOW
2. A Trigger occurs
3. The ATB transaction which contains the Trigger packet is stalled by the CoreSight system for at least 5 cycles

The ATB transaction might be stalled because the trace port cannot output the data quickly enough. For example, if a trace port size of 4 bits is used, the ATB transaction might be delayed for up to 8 cycles. Other trace sources in the system might also affect the available bandwidth on the ATB interface.

Implications

The TRIGOUT signal is asserted multiple times.

This might cause the CoreSight trace sink to indicate multiple triggers in the trace stream or on the trace port.

If the TRIGOUT signal is used in a cross trigger system then multiple trigger events might be signaled to other devices which are connected to the cross trigger system and configured to receive the trigger indication.

This does not affect the ETM's trace stream.

In most trace capture devices, this erratum should not be a problem.

Workaround

A workaround for tools vendors is to only use the first trigger in the trace stream when multiple triggers are present.

Due to the low impact, there is no need to workaround this erratum.

426367: Address comparator compares against incorrect Context-ID

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

If a data transfer from Secure state is presented to the ETM after the processor has changed to Non-Secure state, then any address comparisons performed on this data transfer are compared with the incorrect Context-ID.

Conditions

The following conditions must occur in order:

1. The CoreSight ETM11 is connected to an ARM1176JZ(F)-S processor
2. An Address Comparator is configured for data address comparisons and to match only if a Context-ID comparison also matches (bits[9:8] of the Address Access Type register are not zero)
3. A data transfer instruction is executed in Secure state, but the data takes a long time to return (e.g. a cache miss)
4. The processor changes state to Non-Secure state and executes at least one instruction
5. The data transfer for the Secure data transfer instruction is presented to the ETM which matches the address, access type, size, security state and data comparison for the Address comparator

The Address Comparator can be a Single Address Comparator or an Address Range comparator.

Implications

Under these conditions, the Non-Secure Context-ID is incorrectly used when comparing the data transfer in Condition 5. This means that the comparator might incorrectly match or incorrectly fail to match.

Any events which have been programmed to be sensitive to the Address Comparator might behave unexpectedly. This might include a trigger occurring incorrectly, or trace might be missing or generated unexpectedly.

It should be noted that due to the precise set of conditions, this erratum is unlikely to occur.

Workaround

There is no workaround for this erratum.

426463: Address range comparator behaves incorrectly on unaligned transfer as ProgBit is cleared

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

Under normal conditions, an unaligned transfer is presented to the ETM as two separate but consecutive transfers. If an unaligned transfer occurs just as the programming bit is cleared in the ETM Control Register (0x000) then the range comparator might match incorrectly.

Conditions

1. An Address Range Comparator is set to match on a range of data addresses
2. The unaligned address must be greater than the upper address of the Address Range Comparator
3. The programming bit is cleared just as the unaligned transfer occurs

Implications

The address range comparator might incorrectly match on the unaligned address.

Any events which have been programmed to be sensitive to the Address Range Comparator might behave unexpectedly. This might include a trigger occurring incorrectly, or trace might be missing or generated unexpectedly.

If the ETM is enabled while the processor is halted in debug state then this erratum cannot occur.

Workaround

There is no workaround for this erratum.

426940: Imprecise data abort causes comparator to fire incorrectly

Status

Affects: product CoreSight ETM11.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

Under normal conditions, if a data transfer is imprecisely aborted, address comparators configured to match on data addresses with the Exact Match bit set (bit [7] of the Address Access Type register, 0x020 to 0x02F) should not match.

This erratum causes the comparator to match.

Conditions

The following conditions must exist:

1. An Address Comparator is configured for data address comparisons with the Exact Match bit set
2. A data transfer occurs at the address specified in the Address Comparator
3. The data transfer is imprecisely aborted

This erratum cannot occur when the CoreSight ETM11 is connected to the ARM11 MPCore

Implications

The Address Comparator might incorrectly match on the data transfer. If data value comparisons are disabled (bits [6:5] of the Address Access Type register, 0x020 to 0x02F are zero) then the comparator always matches. If data value comparisons are enabled then it is Unpredictable whether the comparator matches or not.

Any events which have been programmed to be sensitive to the Address Comparator might behave unexpectedly. This might include a trigger occurring incorrectly, or trace might be missing or generated unexpectedly.

Workaround

This is a workaround for tools vendors or users.

The exact match bit should be clear. This means the comparator does not behave Unpredictably, but always matches on the aborted transfer.

Errata - Implementation

435327: PCLKEN input to ETM11CSSingle might cause timing closure failures

Status

Affects: product CoreSight ETM11.

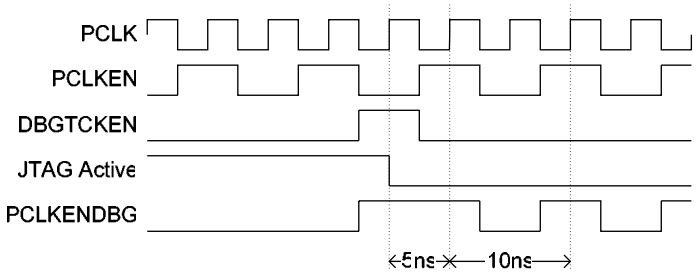
Fault status: Imp, Present in: r0p0,r0p1,r0p2,r1p0, Fixed in r1p1.

Description

This erratum only affects CoreSight ETM11 when used in the ETM11CSSingle configuration.

The ETMJTAGPORT in ETM11CSSingle generates PCLKENDBG as an APB clock-enable for the CoreSight ETM11. PCLKENDBG is driven from either PCLKEN (the system APB clock enable) or DBGTCKEN from the JTAG synchronising logic in ETM11CSSingle. When the ETMJTAGPORT switches from JTAG to system APB or from system APB to JTAG, the PCLKENDBG input to the CoreSight ETM11 might cause consecutive PCLKENDBG pulses to be closer than synthesis constraints allow.

For example, consider if PCLK is driven at 200MHz and PCLKEN is used to produce an effective system APB clock speed of 100MHz. The time between 2 rising PCLK edges where PCLKEN is high is 10ns. If the ETMJTAGPORT switches while PCLKEN is low, then the time between 2 rising PCLK edges where PCLKENDBG is high might be 5ns. This might violate timing constraints. The timing diagram below shows how PCLKENDBG might behave.



Conditions

- The CoreSight ETM11 is used in the ETM11CSSingle configuration
- The system APB uses PCLKEN to give an effective APB speed which is slower than PCLK
- The ETMJTAGPORT is used to program the ETM

Implications

Timing constraints might be violated. This might result in failure to achieve timing closure. This is not expected to cause significant functional problems because the signals which are sampled using PCLK and PCLKENDBG in the CoreSight ETM11 will not be changing around the switching point under normal circumstances.

Workaround

These are 2 workarounds for system implementors.

1. PCLKEN should be tied HIGH. If a slower APB speed is required, PCLK should be gated before being driven into ETM11CSSingle.
2. An external synchroniser should be used to generate DBGTCKEN and DBGTCKEN must be coincident with PCLKEN. JTAGSBYPASS to ETM11CSSingle is tied HIGH. Please contact ARM Ltd for more details.