ARM Core
Cortex-M3 / Cortex-M3 with ETM (AT420/AT425)
Errata Notice

This document contains all errata known at the date of issue in supported releases up to and including revision r2p0 of Cortex-M3 (AT420) and Cortex-M3 with ETM (AT425) products.

Proprietary notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM Limited in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Document confidentiality status

This document is Non Confidential.

Web address

http://www.arm.com/

Feedback on the product

If you have any comments or suggestions about this product, contact your supplier giving:

- The product name
- A concise explanation of your comments.

Feedback on this document

If you have any comments on about this document, please send email to mailto:support-cores@arm.com giving:

- The document title
- The documents number
- The page number(s) to which your comments refer
- A concise explanation of your comments

General suggestion for additions and improvements are also welcome.

Contents

Introduction

Scope

This document describes errata categorised by level of severity. Each description includes:

- a unique defect tracking identifier
- the current status of the defect
- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a 'work-around' where possible

Categorisation of Errata

Errata recorded in this document are split into three levels of severity:

Category 1	Behavior that is impossible to work around and that severely restricts the use of the product in all, or the majority of applications, rendering the device unusable.
Category 2	Behavior that contravenes the specified behavior and that might limit or severely impair the intended use of specified features, but does not render the product unusable in all or the majority of applications.
Category 3	Behavior that was not the originally intended behavior but should not cause any problems in applications.

Change Control

24 Oct 2008: Changes in Document v9

Page	Status	ID	Cat	Summary
39	New	602117	Cat 2	LDRD with base in list may result in incorrect base register when interrupted or faulted

30 Jul 2008: Changes in Document v8

Page	Status	ID	Cat	Summary
38	New	563915	Cat 2	Event Register is not set by interrupts and debug

10 Jun 2008: Changes in Document v7

Page	Status	ID	Cat	Summary
55	New	511864	Cat 3	Cortex-M3 may fetch instructions using incorrect privilege on return from an exception
56	New	532314	Cat 3	DWT CPI counter increments during sleep
57	New	538714	Cat 3	Cortex-M3 TPIU Clock Domain crossing
58	New	548721	Cat 3	Internal write buffer could be active whilst asleep
63	New	531064	Impl	SWJ-DP missing POR reset sync

23 Nov 2007: Changes in Document v6

Page	Status	ID	Cat	Summary
52	New	463763	Cat 3	BKPT in debug monitor mode can cause DFSR mismatch
53	New	463764	Cat 3	Core may freeze for SLEEPONEXIT single instruction ISR
54	New	463769	Cat 3	Unaligned MPU fault during a write may cause the wrong data to be written to a successful first access

13 Nov 2007: Changes in Document v5

Page	Status	ID	Cat	Summary
36	New	429964	Cat 2	Async not generated if no trace in previous session
37	New	429965	Cat 2	Trigger packets sometimes not inserted in trace stream

26 Sep 2006: Changes in Document v4

Page	Status	ID	Cat	Summary
49	New	382860	Cat 3	Halting mode debug stepping can fail to halt on folded IT instruction.
59	New	398784	Cat 3	Serial Wire Output without formatter bypass is unusable if ETM is not present
51	New	404788	Cat 3	Store to external PPB in user mode may not fault precisely
60	New	404810	Impl	Vendor specific section of memory map is permanently little-endian
61	New	404811	Impl	Data phase of instruction fetch may overlap assertion of SLEEPING
62	New	404812	Impl	DBGPWRUP generated incorrectly

11 May 2006: Changes in Document v3

Page	Status	ID	Cat	Summary
33	New	382857	Cat 2	SysTick COUNTFLAG may not get set if core is sleeping
32	New	382856	Cat 2	RXEV signal will not wake WFE if HCLK is suppressed

34	New	382858	Cat 2	Simultaneous priority update and preemption may cause an invalid state to be entered
35	New	382859	Cat 2	Secondary interruption may corrupt ICI continued Thumb-1 LDM with base-in-list.
47	New	377965	Cat 3	ETM uses SE rather than RSTBYPASS to disable clock gate.
46	New	377964	Cat 3	TPIU: The BaudDivRst signal is not synchronised across clock domains
48	New	377985	Cat 3	AHB AP will not respond to DAPABORT if HCLK is stopped
50	New	382862	Cat 3	LR value on entry to handler following use of illegal EXC_RETURN is incorrect

20 Mar 2006: Changes in Document v2

Page	Status	ID	Cat	Summary
16	New	372726	Cat 2	Exit reset and exit debug causes additional E atom to be traced
17	New	375889	Cat 2	De-assertion of NIDEN might cause incorrect trace
18	New	377489	Cat 2	SysTick COUNTFLAG does not get cleared when the SysTick Current Value Register is written.
19	New	377490	Cat 2	SysTick COUNTFLAG automatic clear operation has priority over set.
20	New	377492	Cat 2	Incorrect return PC may be stacked for NMI in parallel with double fault lock up case
21	New	377493	Cat 2	C_MASKINTS in parallel with disabled interrupts can cause local faults to not be taken.
22	New	377494	Cat 2	Instruction incorrectly pre-fetch aborts due to miss-predicted branch
23	New	377496	Cat 2	Non byte size bit-band accesses in BE8 mode access incorrect bits
24	New	377497	Cat 2	Setting C_MASKINTS in the same cycle as halting debug is cleared
25	New	377519	Cat 2	HPROT always reports non-bufferable during HW stacking
26	New	377521	Cat 2	Interrupted alignment-faulting store following failing STREX may corrupt stack.
27	New	377522	Cat 2	SysTick CLKSOURCE is not forced to 1 when SysTick NOREF is high
28	New	377523	Cat 2	Fault on first of a pair of pipelined loads misinforms ETM interface
29	New	377524	Cat 2	Incorrect ETM Peripheral ID
30	New	377525	Cat 2	Authentication Status register reads as 0.
31	New	377681	Cat 2	Interrupted fault-generating load/store pair with SP base-writeback may corrupt stack
42	New	372716	Cat 3	RFE traced after Debug exit
43	New	377491	Cat 3	DWTTRAP/EXTERNAL bits in DFSR can be incorrectly set
45	New	377498	Cat 3	SVC/UNDEF swapped with branch inside HF/NMI lockup
44	New	377495	Cat 3	Debug wake up from SleepOnExit with DHREADY low for PC unstacking address phase

15 Dec 2005: Changes in Document v1

Page	Status	ID	Cat	Summary
14	New	369016	Cat 2	The second or third part of an unaligned transaction may be marked incorrectly

41	New	368884	Cat 3	ETM does not trace a software initiated reset

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	r1p0	r1p1	r1p1-00rel0	r1p1-01rel0	r2p0-00rel0
404810	Impl	Vendor specific section of memory map is permanently little-endian	X	X				
404811	Impl	Data phase of instruction fetch may overlap assertion of SLEEPING	X	X				
404812	Impl	DBGPWRUP generated incorrectly		X				
531064	Impl	SWJ-DP missing POR reset sync			X	X	X	
369016	Cat 2	The second or third part of an unaligned transaction may be marked incorrectly	X					
372726	Cat 2	Exit reset and exit debug causes additional E atom to be traced	X					
375889	Cat 2	De-assertion of NIDEN might cause incorrect trace	X					
377489	Cat 2	SysTick COUNTFLAG does not get cleared when the SysTick Current Value Register is written.	X					
377490	Cat 2	SysTick COUNTFLAG automatic clear operation has priority over set.	X					
377492	Cat 2	Incorrect return PC may be stacked for NMI in parallel with double fault lock up case	X					
377493	Cat 2	C_MASKINTS in parallel with disabled interrupts can cause local faults to not be taken.	X					
377494	Cat 2	Instruction incorrectly pre-fetch aborts due to miss-predicted branch	X					
377496	Cat 2	Non byte size bit-band accesses in BE8 mode access incorrect bits	X					
377497	Cat 2	Setting C_MASKINTS in the same cycle as halting debug is cleared	X					
377519	Cat 2	HPROT always reports non-bufferable during HW stacking	X					
377521	Cat 2	Interrupted alignment-faulting store following failing STREX may corrupt stack.	X					

ID	Cat	Summary of Erratum	r0p0	r1p0	r1p1	r1p1-00rel0	r1p1-01rel0	r2p0-00rel0
377522	Cat 2	SysTick CLKSOURCE is not forced to 1 when SysTick NOREF is high	X					
377523	Cat 2	Fault on first of a pair of pipelined loads misinforms ETM interface	X					
377524	Cat 2	Incorrect ETM Peripheral ID	X					
377525	Cat 2	Authentication Status register reads as 0.	X					
377681	Cat 2	Interrupted fault-generating load/store pair with SP base-writeback may corrupt stack	X					
382856	Cat 2	RXEV signal will not wake WFE if HCLK is suppressed	X					
382857	Cat 2	SysTick COUNTFLAG may not get set if core is sleeping	X					
382858	Cat 2	Simultaneous priority update and preemption may cause an invalid state to be entered	X					
382859	Cat 2	Secondary interruption may corrupt ICI continued Thumb-1 LDM with base-in-list.	X					
429964	Cat 2	Async not generated if no trace in previous session	X	X	X	X		
429965	Cat 2	Trigger packets sometimes not inserted in trace stream	X	X	X	X		
563915	Cat 2	Event Register is not set by interrupts and debug	X	X	X	X	X	X
602117	Cat 2	LDRD with base in list may result in incorrect base register when interrupted or faulted	X	X	X	X	X	X
368884	Cat 3	ETM does not trace a software initiated reset	X					
372716	Cat 3	RFE traced after Debug exit	X					
377491	Cat 3	DWTTRAP/EXTERNAL bits in DFSR can be incorrectly set	X					
377495	Cat 3	Debug wake up from SleepOnExit with DHREADY low for PC unstacking address phase	X					
377498	Cat 3	SVC/UNDEF swapped with branch inside HF/NMI lockup	X					
377964	Cat 3	TPIU: The BaudDivRst signal is not synchronised across clock domains	X					
377965	Cat 3	ETM uses SE rather than RSTBYPASS to disable clock gate.	X					
377985	Cat 3	AHB AP will not respond to DAPABORT if HCLK is stopped	X					

ID	**Cat**	**Summary of Erratum**	**r0p0**	**r1p0**	**r1p1**	**r1p1-00rel0**	**r1p1-01rel0**	**r2p0-00rel0**
382860	Cat 3	Halting mode debug stepping can fail to halt on folded IT instruction.	X					
382862	Cat 3	LR value on entry to handler following use of illegal EXC_RETURN is incorrect	X					
404788	Cat 3	Store to external PPB in user mode may not fault precisely	X	X				
463763	Cat 3	BKPT in debug monitor mode can cause DFSR mismatch	X	X	X	X	X	
463764	Cat 3	Core may freeze for SLEEPONEXIT single instruction ISR	X	X	X	X	X	
463769	Cat 3	Unaligned MPU fault during a write may cause the wrong data to be written to a successful first access	X	X	X	X	X	
511864	Cat 3	Cortex-M3 may fetch instructions using incorrect privilege on return from an exception	X	X	X	X	X	
532314	Cat 3	DWT CPI counter increments during sleep	X	X	X	X	X	
538714	Cat 3	Cortex-M3 TPIU Clock Domain crossing	X	X	X	X	X	
548721	Cat 3	Internal write buffer could be active whilst asleep	X	X	X	X	X	

Errata - Category 1

There are no Errata in this Category

Errata - Category 2

369016: The second or third part of an unaligned transaction may be marked incorrectly

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

For a core issued unaligned transaction, Cortex-M3 converts the transaction into two or three size aligned AHB transactions, as described in section 14.9 of the TRM.

In a Cortex-M3 system where the MPU is not implemented, the values of HPROTD[3:2], MEMATTRD[1:0], HPROTS[3:2] and MEMATTRS[1:0] may have the incorrect values for the second or third transaction resulting from a word-boundary crossing unaligned data transaction from the core. HPROTx[1:0], transaction privilege and opcode vs data identifier, are not affected.

This bug may, for example, result in the second half of an unaligned cacheable transaction being marked as non-cacheable by a Cortex-M3 implementation that does not include an MPU.

Conditions

1. A Cortex-M3 without MPU is implemented
2. An unaligned data transaction that crosses a word boundary is performed

Implications

An external device which bases its handling of a data transaction upon the values of one or more of HPROTD[3:2], MEMATTRD[1:0], HPROTS[3:2] or MEMATTRS[1:0] may perform an incorrect operation due to incorrect information provided by Cortex-M3. For example, a level-2 cache controller may return stale data as a result of having not cached data that should have been marked cacheable by Cortex-M3.

Workaround

Several work arounds exist, both in hardware and software.

1. Implement a Cortex-M3 with the MPU.
2. Ensure that software only performs aligned transactions, for example, using a compiler option.
3. Kernel software may prevent thread code from performing such transactions by setting the UNALIGN_TRP bit in the Configuration Control Register.
4. The memory attributes for a particular address are fixed for a Cortex-M3 implementation without an MPU. Therefore, a system designer may choose either to ignore HPROTx[3:2] and MEMATTRx[1:0], or regenerate them at the system level using HADDRx[31:29] as follows:

```
// Data Code Bus Signals
HPROTD[3:2]   = 2'b10;
```

```
MEMATTRD[1:0] = 2'b01;
// System Bus Signals
HPROTS[3]     = HADDRS[31:29] == 3'b011
              | HADDRS[31:29] == 3'b100;
HPROTS[2]     = HADDRS[30:29] != 2'b00;
MEMATTRS[1]   = HADDRS[31:29] == 3'b101;
MEMATTRS[0]   = ~HADDRS[31] &  HADDRS[29]
              | ~HADDRS[30] & ~HADDRS[29];
```

372726: Exit reset and exit debug causes additional E atom to be traced

Status

Affects: product Cortex-M3 with ETM.

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

Description

The ETM is designed to trace exceptions which occur whilst the core is in debug state, indicating that trace has re-started, followed by an exception branch packet. If the exception is a reset, and an NMI occurs before the first instruction of the reset vector is executed, then two branch packets will be traced back to back. The E atom which is intended to be output between the two exception packets is traced after the 2nd branch.

The trace stream generated is legal, but does not correctly indicate the execution stream. Although the resulting trace is legally formatted, it is unlikely to decompress correctly until the next indirect branch packet is traced.

Conditions

1. Tracing is enabled
2. Core enters debug state
3. Reset occurs
4. NMI occurs
5. Core leaves debug state

Implications

The interpretation of the trace stream which is output is to discard the first exception packet (which indicates reset) and replace the branch with the 2nd exception packet (which indicates NMI). This hides the occurrence of the reset exception. Since the E atom which was intended to separate the two exceptions is output after the 2nd branch packet, an extra instruction will appear in the trace for the NMI handler and this will affect the decompression of the instructions up to the next indirect branch.

Workaround

The ETM will only output back to back exception packets in this specific scenario, there is no intentional generation of branch packets which are then replaced by a new branch packet. Whenever 2 back to back exception branches are observed it can be deduced that the 1st E atom from the next P-header should be removed and inserted between the two exception packets.

375889: De-assertion of NIDEN might cause incorrect trace

Status

Affects:	product Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

The inputs signal NIDEN is provided on the ETM as a global control signal to allow the ETM to operate.

If this signal is LOW, then the ETM should stop tracing and output all trace currently in the FIFO. If NIDEN is subsequently driven HIGH, the ETM should restart tracing at the following instruction boundary.

If this erratum occurs, the ETM FIFO does not empty when NIDEN is driven LOW. When NIDEN 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 ETM is enabled and generating trace
2. NIDEN is driven LOW then NIDEN is driven HIGH

Implications

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

It is not expected that NIDEN will be dynamically changed during tracing since the normal usage model is for NIDEN to permanently disable tracing on a device.

If NIDEN is not dynamically changed during tracing, this erratum does not occur.

Workaround

There is no workaround for this erratum.

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

377489: SysTick COUNTFLAG does not get cleared when the SysTick Current Value Register is written.

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

The ARMv7M architecture states that the COUNTFLAG bit in the SysTick Control and Status Register (0xE000E010) should be automatically cleared when the SysTick Current Value Register (0xE000E018) is written to. In CortexM3 r0p0, COUNTFLAG is not automatically cleared when the SysTick Current Value Register is written to.

Conditions

1. The SysTick COUNTFLAG is currently set.
2. A write to the SysTick Current Value Register is performed.

Implications

Code performing write operations to the SysTick Current Value Register and later relying on the value of COUNTFLAG to determine if the SysTick time period has elapsed may function incorrectly as it may appear that the SysTick period has elapsed when in fact it has not. This does not affect the SysTick counter’s ability to produce reliable interrupts, nor does it affect any other aspect of the SysTick capabilities.

Workaround

To zero the COUNTFLAG, a read may be performed to the SysTick Control and Status Register. The behaviour of the COUNTFLAG being cleared by a write to the SysTick Current Value Register may therefore be emulated by performing a read of the SysTick Control and Status Register immediately after performing a write to the SysTick Current Value Register.

377490: SysTick COUNTFLAG automatic clear operation has priority over set.

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

The ARMv7M specification states that COUNTFLAG, which is present in the SysTick Control and Status register (0xE000E010), is automatically cleared by the hardware when read by software, and is set by the hardware when the SysTick Current Value register (0xE000E018) decrements to zero.

When the SysTick Control and Status register (0xE000E010) is read in the same cycle as the SysTick Current Value register (0xE000E018) decrements to zero, the read operation is erroneously given priority over the tick to zero i.e. clear has priority over set. This means that a tick to zero setting of COUNTFLAG may be lost.

Conditions

1. A read of the SysTick Control and Status register is performed.
2. The SysTick Current Value register decrements from one to zero in the same cycle

Implications

The intended use of this register is to determine that a SysTick handler has extended beyond the SysTick period; however, it is conceivable that the COUNTFLAG may be used as a polling device. If the COUNTFLAG is polled in the exact same cycle as the SysTick decrements to zero, then the COUNTFLAG may be observed never to become set. At the extreme end, code that performs “while(!COUNTFLAG);” may become deadlocked. Note that this erratum does not affect the normal interrupt generating operation of SysTick.

Workaround

If the frequency of polling COUNTFLAG is low, and exact once-per-set operation is not required, then this erratum may have low or zero impact. If a wait on COUNTFLAG is required, then one of the following workarounds may be applied:

1. Use the SysTick handler to implement a volatile variable equivalent to the COUNTFLAG.
2. If raising the core priority is acceptable then polling the PENDSTSET bit may be used.
3. Compare the CURRENT value in the SysTick Current Value register with its previous value and implement a COUNTFLAG like functionality based on the current value being greater than the last value read.

377492: Incorrect return PC may be stacked for NMI in parallel with double fault lock up case

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

If an NMI occurs in parallel to a double-fault then the stacked PC return address will erroneously be set to the instruction after the fault. For example, if an undefined instruction occurs in the HardFault handler and an NMI is asserted in the same cycle as the undefined is executed, then the stacked PC will point to the instruction after the undefined instruction allowing the core to subsequently escape from lockup without corrective action having occurred.

Conditions

1. Code is being executed within the HardFault handler.
2. An instruction which causes a fault is executed.
3. NMI is asserted in the same cycle as the fault is executed.

Implications

This erratum only effects the operation of the core once it has encountered an unrecoverable exception case. Whilst the behaviour of lockup is architecturally defined, the encountering of a lockup condition is expected to be terminal. This erratum results in it not being possible to rely on the core remaining in a lockup scenario; additional care should be taken in factoring the external LOCKUP signal into any watchdog logic.

Workaround

None; do not rely on remaining in lockup as the result of executing an undefined instruction inside the HardFault handler.

377493: C_MASKINTS in parallel with disabled interrupts can cause local faults to not be taken.

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

If C_MASKINTS is set in the Debug Halting Control and Status register (0xE000EDF0) and then an enabled IRQ interrupt is set it will not be taken when halting debug is released, it will just pend (as is the intended behaviour). If a local fault then occurs which is higher priority than what is currently active but lower priority than the pending IRQ it will not activate. The fault should not escalate to HardFault but it should have been taken, instead it will just pend because the C_MASKINTS disabled IRQ is blocking it. This can cause the core to sit idle if the local fault for example was a bus fault or an undefined instruction.

Conditions

1. C_MASKINTS and C_DEBUGEN are set but C_HALT is not. An enabled IRQ is then pended which is masked by C_MASKINTS is followed by a fault. The priority of the fault handler is higher than the current priority but lower than the pending IRQ and it is enabled, that is it won't escalate to HardFault.

Implications

The erratum effects debug operation only. The core could sit idle when it should have been processing a fault. The core will in affect be locked up.

Workaround

Use the pre-emption priorities of the interrupts/faults concerned to prevent this situation or do not use C_MASKINTS in these circumstances. Alternatively, set PriMask or disable the interrupt concerned.

377494: Instruction incorrectly pre-fetch aborts due to miss-predicted branch

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

Using the following code sequence as an example:

```
Address: 8fc2f3f4    Instruction: LDREXH r14,[r11]
Address: 8fc2f3f8    Instruction: LDRSB r12, [r13, #0xB1]
```

Read from address 0xfffffffc generates an error response from instruction memory

```
Address: 8fc2f3fc    Instruction: MOV r15,r14
```

The LDREXH to r14 changes r14 from 0xFFFFFFFD to 0xA730. However, the MOV PC has already been fetched and, predicting that the PC will become 0xFFFFFFFC, the core starts to fetch from 0xFFFFFFFC. This address pre-fetch aborts and then the MOV gets executed. The core associates the pre-fetch abort with the new value for r14 and re-writes the PC as 0xA730. A load/store single instruction or a not taken opcode is needed between the LDR r14 and the MOV PC, such as:

```
LDR R14,[]
LDR Rx,[]    // pipelined against preceding load MOV PC,LR
MOV r15, r14 // or BX LR
```

...or...

```
ITE
LDR R14,[]
!CC opcode
BX LR
```

Conditions

1. A BX or MOV PC branch target is altered two instructions before the branch is executed.
2. A load/store single instruction or a not taken opcode is needed between the LDR r14 and the MOV PC.
3. The original destination must pre-fetch abort.

Implications

The bus fault or HardFault handler could be executed even though a real pre-fetch abort did not occur.

Workaround

Do not have an instruction between the load to the branch target and the branch itself if the value of what would have been the branch target could pre-fetch abort.

377496: Non byte size bit-band accesses in BE8 mode access incorrect bits

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

Bit-band accesses in BE8 mode only function correctly for byte size accesses. Half-word and word transactions access the incorrect byte.

Conditions

1. BE8 pin tied high.
2. A half-word or word size access to the bit-band region is performed.

Implications

Bit-band operations will access, and alter, the incorrect bytes.

Workaround

Perform BE8 bit-band accesses as bytes.

377497: Setting C_MASKINTS in the same cycle as halting debug is cleared

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

If C_MASKINTS in the Debug Halting Control and Status register (0xE000EDF0) is set in the same cycle as C_HALT in the same register is changed from 1 to 0 and if an IRQ is pending and would have been taken as halt was cleared then it can cause the core and NVIC to get into an unknown state.

Conditions

1. The debugger writes to the Debug Halting Control and Status register whilst the core is halted.
2. The write changes C_MASKINTS from 0 to 1.
3. The write changes C_HALT from 1 to 0.
4. An interrupt is pending that would be taken if C_MASKINTS is not set.

Implications

This erratum only affects debug operation. The core and NVIC may get into an unknown state.

Workaround

The ARMv7M specification states that C_MASKINTS can only be written if the core is halted. Since the write effectively occurs whilst halt is cleared this means it is not in halted debug. C_MASKINTS must be written whilst C_HALT is high so first set C_MASKINTS in one write and then clear C_HALT in a subsequent write if that is what is required.

377519: HPROT always reports non-bufferable during HW stacking

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

Whilst performing hardware stacking operations, the core overrides Cortex-M3's internal write buffers in both the core itself, and in the bus-matrix. This information is also displayed via the AHB HPROT signals, which may cause unpredictable results when subsequently reading a stacked parameter from inside an exception, or when writing a new value to the stack for restoration on the way out of an exception.

Conditions

1. Hardware stacking or unstacking occurs
2. The stack is currently in a bufferable region

Implications

This erratum only affects the case where the stack operation is required to be bufferable. The parameters read, or the values written may not be correct reflections of the values preserved or restored by the hardware exception stacking mechanism.

Workaround

Do not expect stacking operations to ever be marked as bufferable

377521: Interrupted alignment-faulting store following failing STREX may corrupt stack.

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

In the following set of circumstances the stack frame for an interrupt may go to the incorrect location:

SP relative operation such as interrupt unstacking

STREX rx,ry,[rz] // local monitor fails so no ahb transaction

Load or store which alignment faults and another interrupt occurs in parallel

Conditions

1. A stack pointer operation is followed by a STREX
2. The STREX fails the local monitor check
3. The following instruction is a load or a store
4. The load/store unalignment faults
5. UNALGN_TRP in the Configuration Control register (0xE000ED14) is set to 1
6. An interrupt unrelated to the fault occurs in parallel.

Implications

The stack frame for the interrupt being serviced is not at the address indicated by the stack pointer.

Workaround

STREX is not automatically generated by the compiler, when inserting STREX make sure it is not followed by a load or store that could alignment fault. Alternatively, keep UNALGN_TRP bit in the Configuration Control register (0xE000ED14) as 0.

377522: SysTick CLKSOURCE is not forced to 1 when SysTick NOREF is high

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

NOREF which is present in the SysTick Calibration Value register (0xE000E01C) states that if it is asserted then the CLKSOURCE bit of the SysTick Control and Status register (0xE000E010) must be forced to 1 and cannot be set to 0.

However, in CortexM3 r0p0 the CLKSOURCE bit is not forced to 1, NOREF has no affect on CLKSOURCE.

Conditions

1. NOREF is tied high (pin STCALIB[25])
2. CLKSOURCE is left as 0

Implications

The SysTick will incorrectly count using the reference clock even though NOREF (pin STCALIB[25]) is tied high. CLKSOURCE can also be read as 0 but it is supposed to be forced to 1.

Workaround

Set CLKSOURCE in the SysTick Control and Status register (0xE000E010) to 1 if the core clock is to be used instead of the reference clock.

377523: Fault on first of a pair of pipelined loads misinforms ETM interface

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

A pair of LDRs may be pipelined back to back by the core. The core may issue an ETMIVALID on the ETM interface for the second LDR before the HRESP from the first LDR has returned. If the HRESP for the first LDR reports a fault, then the ETM interface has insufficient resource to indicate which of the previous two instructions caused the fault.

In this case, the ETM should not trace the 2nd instruction, since it is the first instruction which is interrupted by the exception, and should be flagged as CANCELLED in the ETM trace stream.

The same behavior can be observed when an LDR is followed by an instruction which fails its condition codes in an IT block, or an LDR which is followed by a NOP.

Conditions

1. Tracing is enabled
2. An exception occurs after a single load or store, which is followed by another load/store, a NOP or an instruction which fails it’s condition code.

Implications

This erratum only affects debug operation. The ETM will trace the two instructions, even if the first instruction results in an exception. It is not possible to determine which of the two instructions caused the exception from inspecting the ETM trace. The trace stream will indicate that 2nd of the two instructions is the one which is being executed when the exception occurs. The PC which is stacked will point to the instruction which caused the exception, and the return from the handler will correctly output a branch to the instruction which is cancelled by the exception.

This will affect DWT PC match, causing the second LDR to be identified as having been executed.

Workaround

If ETM tracing is continually enabled until the fault handler returns, the return address will usually be to the first LDR. This indicates the first LDR caused the fault, not the second LDR.

Alternatively, halting debug can be used to examine the return PC which was stacked to determine the instruction which caused the fault.

377524: Incorrect ETM Peripheral ID

Status

Affects:	product Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

PeriphID0 to PeriphID3 registers at addresses 0xE0041FE0, 0xE0041FE4, 0xE0041FE8, 0xE0041FEC all read as zero.

Conditions

1. All conditions.

Implications

Tools are unable to positively identify the ETM.

Workaround

Since the memory map address of the ETM is determined by the ARMv7-M architecture, it is possible to infer that if a peripheral is present with a Component ID of 0xB105900D at addresses 0xE0041FF0 through 0xE0041FFC, with a Peripheral ID registers 0-3 of 0x00000000 with then it is an r0p0 ETM.

377525: Authentication Status register reads as 0.

Status

Affects: product Cortex-M3 with ETM.

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

Description

CM3 ETM Authentication Status register at address 0xE0041FB8 always reads as 0.

Conditions

1. All conditions.

Implications

Tools are unable to determine that the ETM is able to generate trace (NIDEN input should be reflected in this register to indicate if non-invasive debug is enabled).

Workaround

There is no tools based workaround for this erratum

377681: Interrupted fault-generating load/store pair with SP base-writeback may corrupt stack

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

A load/store single instruction using SP as the base register with writeback, which is followed by another load/store single instruction e.g.:

```
LDR Rw, [SP, #0x10]!
STR Rx, [Ry, Rz/#imm]
```

If this sequence encounters a fault during the data phase of the first instruction (i.e. the address phase of the second instruction) in parallel with an interrupt may cause the exception stack frame to be pushed to the wrong offset. This requires that the first load/store return an external bus fault or the second perform an unaligned transaction whilst UNALIGN_TRP is set.

Conditions

1. Load/store single with SP as the base with write back is followed by another load/store single.
2. The first instruction bus faults or the second instruction alignment faults.
3. An asynchronous interrupt occurs in parallel.
4. The system priority is such that both the interrupt may be taken.

Implications

In the unlikely event of encountering this, the erratum will produce the non-recoverable scenario of entering a Handler with a correct SP, but with the stack frame at the incorrect offset.

Workaround

It is not anticipated that external bus faults will be used as a recoverable faulting mechanism on the stack, and as such, whilst feasible, the bus faulting scenario is not believed to pose a real issue.

The alignment fault caused by the second instruction may be suppressed by not enabling UNALIGN_TRP (the default option), or by not producing code where potentially unaligned load/store singles follow stack access load/store singles with base writeback.

382856: RXEV signal will not wake WFE if HCLK is suppressed

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

The WFE instruction may be used to put the core to sleep. The RXEV input may be used to wake the core if sleeping on a WFE instruction. Cortex-M3 implements two clock inputs; a free running clock FCLK, and a sleep gateable clock HCLK. The defect samples the RXEV input using HCLK rather than FCLK. This results in the core being unable to be woken by RXEV if HCLK is suppressed during sleep.

Conditions

1. Cortex-M3 is sleeping due to WFE.
2. HCLK has been stopped as a result of sleeping.
3. The RXEV input is asserted.

Implications

Systems implementing HCLK gating using the SLEEP or DEEPSLEEP outputs of Cortex-M3 may not wake up in response to an RXEV.

Workaround

Care must be taken when implementing external HCLK gating if a WFE and RXEV are to be used for sleep and wake. An alternative implementation would be to use an interrupt pin and SEVONPEND. A second alternative would be to implement an external FCLK registration flop for RXEV allowing the pulse to be held until an HCLK occurs.

382857: SysTick COUNTFLAG may not get set if core is sleeping

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0, Fixed in r1p0.

Description

The SysTick periodic counter inside the Cortex-M3 design may be clocked from either of the FCLK or STCLK inputs. Both inputs are resynchronized to FCLK before being used to update the SysTick current value register. Transition from one to zero of the SysTick current value should cause the SysTick COUNTFLAG to be set. If the HCLK input to the core is gated (for example in response to the core sleeping), then updates of the SysTick COUNTFLAG are inhibited. This may erroneously result in the loss of a SysTick transition to zero COUNTFLAG update.

Conditions

1. HCLK is externally gated.
2. SysTick Current Value register decrements to zero.

Implications

A SysTick count-to-zero may fail to be recorded in COUNTFLAG. This defect does not affect the SysTick interrupt's ability to wake the core from sleep.

Workaround

Do not rely on the value of COUNTFLAG after sleeping.

382858: Simultaneous priority update and preemption may cause an invalid state to be entered

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

When an interrupt occurs and priority allows, Cortex-M3 should preempt to the associated handler and set all internal and NVIC state accordingly. If two or more interrupts occur in parallel and current priority would allow either interrupt to preempt, Cortex-M3 should preempt to the highest priority handler. If in the same cycle as the preemption occurs, the priority of the interrupts is inverted by software (e.g. the highest priority interrupt is made the lowest priority) then Cortex-M3 may erroneously enter the handler for the original highest priority interrupt but the NVIC will be updated for the new highest priority interrupt. This results in a non-recoverable discontinuity between the NVIC and core states.

Conditions

1. Two interrupts are pending and enabled.
2. Both are higher priority than current priority.
3. The priority of the higher is dropped before the core completes preemption.

Implications

Generating this scenario will result in the core entering an UNPREDICTABLE state. Exiting the handler entered will generate an INVPC (exception not active) exit-fault exception. NMI and HardFault routines will still be serviced correctly.

Workaround

Do not invert the priority of an enabled interrupt with respect to other enabled interrupts when said interrupts could preempt the current priority. This may be achieved by only changing the priority from a higher/equal priority handler, by setting PRIMASK, or by temporarily disabling the interrupt before changing its priority.

382859: Secondary interruption may corrupt ICI continued Thumb-1 LDM with base-in-list.

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

Cortex-M3 allows the interruption and continuation of load-multiple (LDM) instructions. The continuation is implemented by pushing interrupt-continuation-information (ICI) onto the stack at interrupt preemption time. The register from which the LDM is resumed is then a function of the ICI bits. Cortex-M3 implements the base-restore exception model. This means that if an exception occurs, any modification that has been performed to the base register must be unwound. In this situation, Cortex-M3 must temporarily buffer the original base register value in case of exception. In the case of a return to a 16-bit Thumb-1 LDM with base-in-list and ICI bits set (in general as the result of an earlier preemption of the same LDM), the value of the temporary buffer is erroneously not re-updated to the base register value. If a subsequent exception occurs after the base has been loaded from the list, the base restore mechanism may update the base with an UNKNOWN value. A later exception return to this instruction will continue using an UNKNOWN base address.

Conditions

1. 16-bit Thumb-1 LDM with base-in-list at positions other than first or last The LDM is returned to with valid non-zero ICI bits. The LDM continues past where the base is written from the list. An exception/interrupt occurs after the base in list has been loaded

Implications

Code containing Thumb-1 LDM with base-in-list, where the register is not the first or the last in the list, may execute incorrectly if the instruction could be preempted.

Workaround

Avoid the use of Thumb-1 LDMs with base-in-list at positions other than first or last in list. If not possible, then interrupt handlers should clear the stacked ICI bits to zero if they do not describe an IT combination - this will cause the LDM to restart rather than continue.

429964: Async not generated if no trace in previous session

Status

Affects:	product Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0,r1p0,r1p1,r1p1-00rel0, Fixed in r1p1-01rel0.

Description

The ETM is required to generate an alignment synchronisation packet as the first packet every time the programming bit is cleared, in order to allow the tools to synchronise with the protocol stream.

If no trace is generated in a session, the A-sync packet will be correctly generated as the progbit is cleared, but if the progbit is then set and cleared again, second and subsequent A-sync packets are not generated. No other aspects of the trace generation are affected by this, and once any other trace packet is generated the A-sync logic will be correctly reset.

Conditions

This erratum only occurs when the following sequence of events occurs:

1. The ETM is programmed and enabled
2. No instructions are traced
3. The ETM programming bit is set
4. The ETM programming bit is cleared.

The absence of any traced instructions can be due either to the Trace Enable conditions not being satisfied, or the core not executing any instructions.

Implications

This erratum only affects the trace stream which is generated by the ETM. There is no impact on the normal processing operation of the core.

The conditions which cause this erratum are expected to occur whilst debug is being performed.

The A-sync packet may be missing at the start of a trace session. All trace packets which are generated will still be correct, and there is no corruption of the trace stream. If the A-sync from a previous session was captured, it can be used for synchronisation.

Tools will typically fail to capture the trace relating to any bytes of trace before the next periodic A-sync packet is seen. For short trace sessions, it is possible to lose the whole trace session.

Workaround

There is no tools workaround.

Users may attempt to work around this erratum by forcing the generation of trace packets as part of an initialisation sequence. This is only possible if the core is not in debug state, and so may not be applicable in all cases. With this ETM it is also possible to deduce that the first byte of trace captured after the progbit is cleared will be the start of a packet, provided that the formatter in the TPIU is enabled.

429965: Trigger packets sometimes not inserted in trace stream

Status

Affects:	product Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0,r1p0,r1p1,r1p1-00rel0, Fixed in r1p1-01rel0.

Description

It is possible to configure a trigger event for the ETM which is used to assist with trace capture and the subsequent analysis of trace by the user. The trigger condition is indicated by a pulse on the ETMTRIGOUT signal, and is also inserted in the trace stream using a special packet.

If a trigger condition occurs when there is no data in the ETM's FIFO and there are no instructions yet to be traced, the ETMTRIGOUT signal is pulsed correctly, but the trigger packet is not inserted in the trace stream.

Conditions

1. The ETM is enabled
2. The trace FIFO is empty
3. There are no instructions already executed but not yet entered in the trace fifo.

Implications

This erratum only affects the trace stream which is generated by the ETM. There is no impact on the normal processing operation of the core.

This erratum only affects the inclusion of the trigger packet in the trace stream. It does not affect the visibility of the trigger condition through the ETM's programmers model (bit 2 of the ETM Status Register, register 4).

It does not affect the indication of the trigger condition to the trace capture device, and a formatter trigger packet will be inserted if enabled.

The erratum does not occur if a trigger is generated using DWT to detect an instruction address if Trace Enable is high.

When this erratum occurs, the user will not be able to determine the location in the executed instruction sequence at which the trigger condition occurred. Due to the conditions which are required for this erratum to occur, it is more likely that the erratum occurs when an external input is being used to generate the trigger condition.

Workaround

There is no workaround to force the trigger packet to be included in the trace stream. If using the CoreSight formatter protocol, e.g. TPIU/ETB with trigger embedded, then the position of the embedded trigger can be used, however the location of the embedded trigger is approximate.

563915: Event Register is not set by interrupts and debug

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 2, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0,r2p0-00rel0, Open.

Description

The event register used for WFE wake-up events should be set for the following conditions:

1. Event communication (including SEV on local processor)
2. Any exception entering pending state when SEVONPEND is set
3. Exception entry
4. Exception exit
5. Debug event when debug is enabled

In r0p0, r1p0, r1p1 and r2p0 versions of Cortex-M3 the event register is not set for the exception entry, exception exit or debug events.

Conditions

1. An interrupt or debug event occurs whilst the internal event register is clear and the core is not sleeping
2. A WFE is executed
3. No further interrupts or events occur

Implications

If interrupts related to a WFE sleep can be generated before the WFE is executed then it may be possible for the event to be missed. The interrupt will occur and the handler will be executed for that interrupt but the event register will not be set. When the WFE is executed it will go to sleep and not wake up if no other events or interrupts occur.

Workaround

An implementation time workaround is to connect up the missing events to the RXEV input of Cortex-M3. This can be achieved by decoding interrupt events using ETMINTSTAT as well as using the HALTED output. RXEV needs to be asserted whenever ETMINTSTAT is equal to 3'b001 or 3'b010 or when HALTED is asserted. Since ETMINTSTAT is part of the ETM interface the ETM interface needs to be enabled for this workaround. This is achieved by asserting the ETMPWRUP input on Cortex-M3.

A software workaround is to insert the SEV instruction at the beginning and end of all exception handlers.

602117: LDRD with base in list may result in incorrect base register when interrupted or faulted

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Cat 2, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0,r2p0-00rel0, Open.

Description

LDRD with the base register in the list of the form `LDRD Ra, Rb, [Ra, #imm]` may not complete after the load of the first destination register due to an interrupt before the completion of the second load or due to the second load getting a bus fault or an MPU fault. Since the base register has been updated the base register must be restored to its original value before entering the appropriate interrupt or fault handler so that the instruction can restart correctly upon return from the handler. In certain circumstances this may not occur as required.

When the LDRD is interrupted in between the two loads then the base register may not be restored as required. This can only happen when the instructions are being executed from the system bus (address 0x20000000 and above) and the loaded data is also being read from the system bus.

For the fault case where the second load gets a bus fault or an MPU fault then the base register is never restored and there is no dependence on which bus the instructions are being executed from.

When the base register is the second register in the LDRD list, of the form `LDRD Rb, Ra, [Ra, #imm]`, then this erratum cannot occur.

You will not be affected by this erratum if:

1. you do not execute code from the system bus and if your code bus does not generate bus faults and you do not execute LDRD’s that cross MPU boundaries, or
2. if your compiler does not generate LDRD’s

Conditions

Either:

1. An LDRD is being executed where the base register is in the list and write-back is not used:
   `LDRD Ra, Rb, [Ra, #imm]`
2. Instructions and data are both being fetched via the system bus. This occurs for locations in memory greater than 0x20000000.
3. The first LDRD address is prioritised and issued to the system bus, whilst the instruction fetch is internally waited. The instruction fetch is issued to the system bus upon completion of the first part of the LDRD. The second part of the LDRD is issued to the system bus upon completion of the instruction fetch.
4. An interrupt occurs in between the two load operations.

Or:

1. An LDRD is being executed where the base register is in the list and write-back is not used:
   `LDRD Ra, Rb, [Ra, #imm]`
2. A bus fault or MPU fault occurs for the second load.

Implications

The base register will not be restored as expected preventing the instruction from being restarted correctly upon return from the interrupt service routine or from the fault handler.

Workaround

There are two workarounds for this erratum. However, if the instructions are always executed from the code space and faults cannot occur then a workaround is not required.

The first workaround is to replace the LDRD instruction affected by this erratum with other suitable instructions.

```
LDRD Ra, Rb, [Ra, #imm]
```

may be directly replaced by two LDR instructions which will produce exactly the same functionality:

```
LDR Rb, [Ra, #imm + 4]
LDR Ra, [Ra, #imm]
```

Alternatively, an LDRD with base in list may still be used if the base register is the second register in the list:

```
LDRD Rb, Ra, [Ra, #imm]
```

However, in order to achieve the same functionality this requires that the data at the two addresses are swapped, or that the following instructions using Ra or Rb swap their source registers.

The second workaround can be applied when using Cortex-M3 r2p0. It is possible to prevent this erratum occurring for the interrupt case by setting DISMCYCINT (bit[0]) in the Auxiliary Control Register which is located at address 0xE000E008. This bit prevents the interruption of multi-cycle instructions and will therefore increase the interrupt latency of Cortex-M3.

Setting DISMCYCINT does not prevent the second load being faulted which means that the base will still be incorrect for bus faults or MPU faults.

Due to the performance impact of workaround two, and because workaround two does not address the fault conditions of this erratum, ARM recommends that workaround one is used.

Errata - Category 3

368884: ETM does not trace a software initiated reset

Status

Affects: product Cortex-M3 with ETM.

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

Description

The Cortex-M3 processor can be reset by either:

1. A power-on-reset via the input pin PORESETn
2. A system reset via the input pin SYSRESETn
3. A software generated reset by setting bit[0] of the Application Interrupt and Reset Control Register

All resets, except power-on-reset, should be traced by the ETM as exceptions. However, in the case of a software initiated reset, the Cortex-M3 ETM does not trace it as an exception. Instead, the trace will indicate a branch to the reset handler.

Conditions

1. Tracing is enabled
2. A software initiated reset is applied

Implications

Software initiated resets are not indicated as an exception in the trace stream.

Workaround

There is no tools-based workaround for this erratum

It is possible to detect that a software initiated reset has occurred by checking the vector address of a branch in the trace stream and observing that there is no other cause for the branch to have occurred.

372716: RFE traced after Debug exit

Status

Affects: product Cortex-M3 with ETM.

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

Description

If the core enters debug state after an exception return, and trace is then enabled, the first packet output after the I-Sync packet will be an RFE packet. Additionally, if the core enters debug state immediately after executing a single instruction, there will be no P-header corresponding to the instruction executed.

This erratum does not result in information being lost from the trace stream which cannot be inferred from the trace which is output.

Conditions

1. Tracing is disabled
2. A return from exception instruction is executed
3. Core enters debug state
4. Tracing is enabled and core leaves debug state

Implications

The trace stream generated when these conditions are observed can not be decompressed correctly until the next indirect branch is traced. There will either be an additional RFE packet inserted at the start of the stream, or a single E atom P header will be replaced by an RFE packet.

Workaround

The ETM will only output an RFE packet as the first packet after an Isync packet with a debug reason code as a result of this erratum. In this case, the RFE packet should be discarded, and if there is no P-header in the trace stream before the next non-periodic Isync packet or branch packet, then a single E atom p-header should be inferred in its place.

377491: DWTTRAP/EXTERNAL bits in DFSR can be incorrectly set

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

DWTTRAP/EXTERNAL DFSR (0xE000ED30) can be incorrectly set if the debug monitor ISR is enabled and the debug monitor is manually pended in the same cycle as a watch point or external debug request is asserted where the debug monitor ISR's priority is less than the current active priority (i.e. debug monitor cannot activate).

If the above occurs then the debug monitor pend bit will be set by the manual pend as the pre-emption priorities have no affect on a manual pend operation. However, the priority does affect the DWTTRAP and EXTERNAL bits so, for example, a watch point would not have set the pend bit if the debug monitor's pre-emption priority is not greater than the current ISR's pre-emption priority. This would also mean that it should not set the WATCHPOINT DFSR bit. However, if the debug monitor is manually pended in the same cycle as a WATCHPOINT or EDBGRQ in the above circumstances then the EXTERNAL or DWTTRAP FSR bit may be incorrectly set even though the WATCHPOINT or EDBGRQ had no real affect.

Conditions

1. Debug monitor ISR is enabled.
2. Halting debug is disabled.
3. Debug monitor is manually pended in the same cycle as a watch point or external debug request is asserted.
4. Debug monitor ISR's priority is less than the current active priority (i.e. debug monitor cannot activate).

Implications

This erratum affects debug operation only. The DWTTRAP or EDBGRQ status bit in the DFSR may be incorrectly read as set in some extra-ordinary circumstances. This should be of no serious consequence.

Workaround

Should a workaround be required, do not manually pend the debug monitor if it could occur in parallel to an external debug request or watch point and the debug monitor cannot activate (the pre-emption priorities do not allow it). Alternatively raise the debug monitor pre-emption priority.

377495: Debug wake up from SleepOnExit with DHREADY low for PC unstacking address phase

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

A sleep on exit occurs and after sleeping the core gets a wake up via a debug halt request which causes the core to carry out an unstacking operation before it halts. If a data bus conflict occurs on the first transaction for the PC pop the PC unstacking write doesn't occur and the PC gets put into the CPSR, the rest of the unstacking operations will also be one out. This can only occur with a debug halt after a sleep on exit, i.e. it does not affect WFI/WFE debug wake up or interrupts waking up sleep on exit as is the normal system behaviour.

This situation is very difficult to get into as the DHREADY to the core needs to be low for the first unstacking transaction. Since the core will be idle due to the sleep and the debug wake up will be to PPB which is zero wait-stated it will only be possible by some other system bus conflict.

Conditions

1. SleepOnExit used to put the core to sleep after completing an ISR
2. Halting debug is enabled.
3. A debug halt request occurs via the Debug Halting Control and Status register (0xE000EDF0).
4. DHREADY is low for the address phase of the PC unstacking AHB transaction

Implications

This erratum only affects debug operation. The core will be in an unknown/incorrect state but this situation should never arise since AHB-Lite recommends that HREADY must be high for an address phase without a preceding transaction.

Workaround

Do not use debug to wake up the core from a SleepOnExit sleep if there is a possibility that conflict could occur on the data bus for the first AHB transaction due to the unstacking operation. This will not be needed if the AHB recommendation of zero wait state responses to IDLE transfers is followed.

377498: SVC/UNDEF swapped with branch inside HF/NMI lockup

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

If a faulting instruction is executed inside the HardFault or NMI ISRs CortexM3 will lock up and attempt to redo the instruction. If the faulting instruction in memory is swapped with a branch before it is re-fetched, the branch should then be decoded and executed and then the branch target should be decoded and executed. However, the core does the branch but then decodes another instruction without fetching from the branch target.

Conditions

1. HardFault or NMI ISR is being executed.
2. A faulting instruction occurs.
3. The faulting instruction's encoding is swapped in memory with a branch before the core can re-fetch it.

Implications

This erratum only effects the operation of the core once it has encountered an unrecoverable exception case. Whilst the behaviour of lockup is architecturally defined, the encountering of a lockup condition is expected to be terminal. Attempting to patch an erroneous SVC instruction inside a HardFault or NMI handler with a branch instruction may result in the core entering an unknown state.

Workaround

Do not use faulting instructions inside the HardFault/NMI handlers and if they are used do not try and swap them for valid instructions whilst the core is running.

377964: TPIU: The BaudDivRst signal is not synchronised across clock domains

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

In the TPIU provided with Cortex-M3, the signal BaudDivRst which is the output of a register clocked by HCLK is captured by a register clocked by TRACECLKIN with no dual-flop resynchronisation stage. There is a risk that this can lead to metastability, and consequently incorrect loading of the baud rate PRESCALER in the TRACECLKIN domain. Once the baud rate counter counts down, it will be re-loaded with the correct value.

Conditions

1. Tracing is enabled.
2. TPIU PROTOCOL is set to one of the two Serial Wire Output modes.
3. TRACECLKIN and HCLK are not synchronous

Implications

This erratum only affects the operation of the TPIU when Serial wire output modes are being used. It is normal that writing to PRESCALER register will corrupt data if the write occurs whilst data is being output, so it is likely that no effect will be observed from this erratum. Since the baud rate counter is reloaded with the register value from the HCLK domain every baud period, only the count when a write is made will corrupted.

Workaround

In order to ensure that the PRESCALER register is correctly written, it is possible to perform two writes to this register with the same value. This will ensure that the reload value used by the counter has stabilized by the time the second write causes a reload to occur.

377965: ETM uses SE rather than RSTBYPASS to disable clock gate.

Status

Affects:	product Cortex-M3 with ETM.
Fault status:	Cat 3, Present in: r0p0, Fixed in r1p0.

Description

The Implementation Guide describes the pin RESETBYPASS as being used to disable the architectural clock gating cells which exist in the core for scan testing. The CM3ETM uses the SE pin to perform the same function. Although the use of SE in the ETM will not cause any problems, it is not as described in the documentation, and for consistency the ETM will be changed to use the same mechanism as the CortexM3 block.

Implications

This erratum highlights a discrepancy between the documentation and the implementation, but there should be no impact on the use of the design.

Workaround

No workaround is required for this erratum.

377985: AHB AP will not respond to DAPABORT if HCLK is stopped

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

If a DAP access is performed using the AHB-AP debug port, and HCLK is stopped before the transfer completes, then the debug port will be stalled until HCLK is re-started. The debug port will still respond to transfer requests, but will indicate that there is a transfer in progress. Even if a DAPABORT request is transmitted, the AHB-AP will not respond until HCLK is re-started.

Conditions

1. Debug accesses are being performed
2. HCLK is stopped during the debug session

Implications

In a single core Cortex-M3 system, the impact of this erratum is low. Although it affects the response which is obtained by debug tools which are attempting to perform a debug access with HCLK stopped, the scenario of HCLK being stopped whilst a debug session is in progress will in itself prevent the debug session from continuing. The impact of this erratum is therefore one of visibility rather than functionality. In a system containing multiple access ports, this erratum will prevent debug access to other cores, which may have clocks present and be able to respond to debug transactions.

Workaround

There is no workaround to this erratum.

382860: Halting mode debug stepping can fail to halt on folded IT instruction.

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 3, Present in: r0p0, Fixed in r1p0.

Description

For performance reasons, the Cortex-M3 will attempt to fold IT instructions onto a preceding 16-bit data-processing operation. This means that the IT instruction is merged with the previous execution cycle and is therefore not distinct from the previous instruction. Under halting mode stepping using the C_STEP facility, suppression of this folding takes place to provide a more friendly debug experience. The errata manifests itself such that if the IT is within the same 32-bit word as the preceding data-processing operation, then suppression of the folding may fail. This will result in the core appearing to step two instructions in one go, namely the preceding instruction and its following IT.

Conditions

1. Halting mode debug stepping is being performed via C_STEP. The core is halted on a 16-bit instruction in the lower half of a 32-bit word with an IT instruction in the upper half.
2. The core is stepped.

Implications

Single stepping may appear to skip an IT instruction. There is no functional error with respect to the core's state and in most cases this should just be a distraction whilst debugging.

Workaround

none

382862: LR value on entry to handler following use of illegal EXC_RETURN is incorrect

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

The exception handler entry value of the link-register (LR) following an illegal exit is architected to be the EXC_RETURN value used to generate the illegal exit. In some scenarios, Cortex-M3 may erroneously regenerate the LR value to be a valid EXC_RETURN value. The fault handler entered, and the fault-status flags are still set correctly, and it is only the LR value that is incorrect.

Conditions

1. An illegal exit is performed using an EXC_RETURN that is reserved or not valid for the current core state

Implications

The conditions required to see this defect are only visible through erroneous software, and only by software trying to debug this error. The illegal exit may still be detected as the fault cause via the INVPC flag, however, further qualification via the LR copy of the EXC_RETURN used is not possible.

Workaround

None.

404788: Store to external PPB in user mode may not fault precisely

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

The only faults permissible on the private peripheral bus are due to privilege violations and a non-bufferable store should always fault precisely. Since PPB is always marked as

non-bufferable, all user accesses that cause privilege violations should therefore fault precisely. In Cortex-M3 user accesses that cause privilege violations when storing to the external PPB, may fault imprecisely due to the write-buffer treating them as

bufferable.

Conditions

1. An imprecise store is performed to the external private peripheral bus.
2. The store is a user (un-privileged) access.

Implications

The user store to the external PPB will be treated as bufferable and therefore it could possibly fault imprecisely. This means that the fault address register will not contain the faulting address and more instructions may have been executed since the faulting store has been executed.

Workaround

It is not expected that any code would expect this to be a recoverable fault. User code cannot access the external PPB segment so any store will always fault and should be expected

463763: BKPT in debug monitor mode can cause DFSR mismatch

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

A BKPT may be executed in debug monitor mode which will cause the debug monitor handler to be run but the Debug Fault Status Register (DFSR) at address 0xE000ED30 will not have bit 1 set to indicate the cause was a BKPT instruction. This will only occur if an interrupt other than the Debug Monitor is already being processed just before the BKPT is executed.

Conditions

1. C_DEBUGEN (bit 0) in the Debug Halting Control and Status Register at address 0xE000EDF0 is 0.
2. MON_EN (bit 16) in the Debug Exception and Monitor Control Register at address 0xE000EDFC is 1.
3. An enabled interrupt occurs two cycles before the BKPT is executed that causes a pre-emption.

Implications

The Debug Monitor handler may be entered without the DFSR revealing the cause of the handler being entered.

Workaround

Should a workaround be required, it can be deduced that if the DFSR does not have any bits set when the debug monitor has been entered then the cause must be due to this corner case and that it was the result of a BKPT.

463764: Core may freeze for SLEEPONEXIT single instruction ISR

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

The SLEEPONEXIT functionality causes the core to enter the sleep mode when the exit from the sole active interrupt occurs. This means that there are no more interrupts active and the exit would have caused a return to the thread.

It is possible for the core to become frozen if the SLEEPONEXIT functionality is used and the interrupt service routine (ISR) concerned only contains a single instruction. This freezing may occur if only one interrupt is active and it is pre-empted by an interrupt whose handler only contains the single instruction. This instruction must be a legal ISR exit instruction that takes one cycle to execute (either a BX or a BLX). In this case the unstacking would occur after the single instruction had been executed as normal to return to the now only active interrupt handler. However, once it has returned no more instructions will be processed and the core will be frozen. Any new pre-empting interrupt will unfreeze the processor.

Conditions

1. SLEEPONEXIT (bit 1) in the System Control Register at address 0xE000ED10 is set.
2. An interrupt occurs that causes a pre-emption of the current ISR which is the only interrupt that is currently active.
3. The interrupt service routine that is entered consists of only one instruction (either BX or BLX) which causes a legal exit from that ISR.

Implications

The core may freeze and stop processing instructions when it returns to the only currently active ISR. Note that a new interrupt that causes a pre-emption would cause the core to become unfrozen and behave correctly again.

Workaround

If the SLEEPONEXIT functionality is required then do not allow an ISR to contain only one instruction. If an empty ISR is used then insert a NOP before the exit instruction.

463769: Unaligned MPU fault during a write may cause the wrong data to be written to a successful first access

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

When an unaligned store is executed by Cortex-M3 the transaction is split up into either two or three aligned transactions forming constituent parts of the larger transaction. The MPU will check that these transactions are permitted and will block them if necessary. If an unaligned transaction occurs where it overlaps two MPU regions then each region relating to the part of the transaction that hits that region will be checked.

If an unaligned store occurs that crosses an MPU region boundary and has an MPU permission fault for the second region check but not for the first region then it is possible for the second component's data to be written for the first successful transaction in place of the first transaction's data. This can occur for writes to either the D-Code or system bus but will only occur if one or more wait-states are applied for the first component of the store.

Conditions

1. The full MPU is present and enabled with at least one region programmed and enabled.
2. An unaligned store is executed by the processor. The store can be to either the D-Code or the System bus.
3. The store crosses an MPU region boundary.
4. The first region lookup passes, the second region lookup fails.
5. One or more wait states are applied via HREADYS or HREADYD.

Implications

The wrong data will be stored to a permitted address. However, a MemManage fault will occur immediately pointing to the instruction that caused the fault. This may lead to the instruction being re-executed and the store occurring successfully if it is for non-device memory. This would mean that the previously stored data would be overwritten and the wrong value would never be seen. This may not be true for a shared memory system.

Workaround

A workaround is only required if the MPU is present and enabled. Either:

1. do not allow accesses to span more than one region or
2. do not allow unaligned accesses at all or
3. program the MPU correctly if applicable

511864: Cortex-M3 may fetch instructions using incorrect privilege on return from an exception

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

Whilst unstacking registers on return from an exception to a User-privilege thread, Cortex-M3 attempts to simultaneously prefetch the thread's instruction stream. Before the register unstacking is complete, upto the first three memory transactions used to perform instruction prefetching may be erroneously marked as Privileged. This may allow between three and six instructions from a Privileged-access-only region to be executed by a User-privilege thread.

Once fetched, the instructions are executed with User-privilege. Instruction fetches performed after register unstacking has completed will be performed with User-privilege. Both the register unstacking, and any data-transactions generated by executing the erroneously executed instructions will be performed and correctly marked as User-privilege.

Conditions

1. Exception return is executed
2. The exception return is to user code

Implications

User-privileged code may contrive a situation in order to allow execution of up to three words worth of instructions intended to be accessible to Privileged-only execution; however, execution of said instructions will always be performed with User-privilege, thus there are no additional capabilities provided to User-privilege through this erratum.

There exists a theoretical possibility that User-privilege code could use this erratum to allow limited extraction of code and or data from Privileged-access only memory.

Note that read sensitive Privileged-access only peripherals should always be placed in an XN region either via the default memory map, or via the optional memory-protection-unit. Alternatively such peripherals should ignore transactions with HPROT[0] indicating that the transaction is an instruction fetch.

Workaround

None.

532314: DWT CPI counter increments during sleep

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

The DWT contains a number of counters for the profiling of applications. The CPI counter is used to indicate the total number of clock cycles beyond the first cycle of each instruction. The counter is specified to not increment whilst the core is sleeping but for previous revisions it does increment. This results in sleep cycles being counted as program execution cycles.

Conditions

1. The CPI counter in the DWT is enabled
2. Core sleeps during profiling

Implications

Profiling information could be calculated incorrectly if the following calculation is used:

InstructionCount = CycleCount – (CPIcount+LSUcount+INTcount+SLEEPcount) + FOLDcount

Workaround

The number of sleep cycles given by SLEEPCNT can be subtracted from the CPI cycle count to obtain the correct CPI cycle information. Use the following equation:

InstructionCount = CycleCount – (CPIcount+LSUcount+INTcount) + FOLDcount

538714: Cortex-M3 TPIU Clock Domain crossing

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

Combinatorial paths exist in control signals crossing the asynchronous clock boundary between FCLK and TRACECLKIN.

Some of these signals control the reading and writing of data in the trace data FIFO on both sides of the FCLK and TRACECLKIN clock boundary and therefore could cause old data to be repeated or new data to be lost.

Conditions

1. FCLK/HCLK is asynchronous to TRACECLKIN

Implications

When FCLK and TRACECLKIN are asynchronous and depending on the silicon implementation of the block, trace data might become corrupted.

Workaround

This is a workaround for system implementers. System implementers should make FCLK and TRACECLKIN operate synchronously. To avoid the possibility of corrupted trace data, the Trace Port must be fed with a clock synchronous to FCLK. Any crossing to an asynchronous TRACECLKIN domain should be done externally before the TPIU via a separate ATB asynchronous bridge.

548721: Internal write buffer could be active whilst asleep

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Cat 3, Present in: r0p0,r1p0,r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

If a store immediate that is marked as not strongly ordered is used immediately before a WFE or WFI then the store may still be in progress when the core has asserted the SLEEPING signal. This will only occur if wait-states are applied to the store operation. This will not cause a problem unless the location that the store was accessing was using a free-running clock whilst a clock-gating cell has been used to gate FCLK to form HCLK.

Conditions

1. A store with immediate offset is executed
2. The store operation is allowed to be bufferable
3. The store is followed immediately by a WFI or WFE
4. Wait-states are used to delay the data-phase of the store
5. A clock-gate is used to produce HCLK which is gated when SLEEPING is asserted
6. The location the store was to is using a free-running version of the core clock (FCLK)

Implications

An imprecise error response could be missed if it is issued by the peripheral whilst the core is asleep when the peripheral is not using a gated clock but the core is. The stored data will be correct as the core will hold HWDATA at the correct value until it wakes from sleep and completes the transaction.

Workaround

A software workaround is to insert DSB instructions before any WFE and WFI instructions in the application code.

398784: Serial Wire Output without formatter bypass is unusable if ETM is not present

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

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

Description

This erratum relates to the TPIU inside the CortexM3Integration layer. It is not possible to generate Serial Wire Output with the formatter disabled when used in the non-ETM configuration. Trace data will be accepted by the TPIU, but the data will not be passed through to the output pins. The formatter is enabled by default, to bypass the formatter bit 1 (EnFCont) in the Formatter & Flush Control Register at address 0xE0040304 needs to be cleared.

Conditions

1. ETM is not present in the implementation
2. Serial wire protocol (Manchester or NRZ) is selected.
3. Formatter bypass is used

Implications

Although being able to bypass the formatter would decease the bandwidth needed for trace output. The bandwidth saved is not significant.

If the formatter bypass is used then the data output from the ITM or the DWT (via the ITM) will be lost.

Workaround

When using the Serial Wire Output feature on designs where the TPIU is configured for use without the ETM do not bypass the TPIU formatter.

Errata – Implementation

404810: Vendor specific section of memory map is permanently little-endian

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Impl, Present in: r0p0,r1p0, Fixed in r1p1.

Description

The vendor-specific area in the ARMv7-M memory map (from 0xE0100000 to 0xFFFFFFFF) should be affected by the BIGEND pin but it is permanently fixed at little-endian. The private peripheral bus section is the only section in the memory map that should be

unaffected by the BIGEND pin.

Conditions

1. BIGEND is tied high.
2. An access is made to the vendor-specific area of the memory map

Implications

The data read from or written to the vendor-specific area will be in little-endian format instead of the expected BE8 format.

Workaround

A workaround is only required for BE8 systems. The little-endian to BE8 format conversion can be performed externally to Cortex-M3 for the vendor-specific area based upon HADDR and HSIZE for the transaction.

404811: Data phase of instruction fetch may overlap assertion of SLEEPING

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Impl, Present in: r0p0,r1p0, Fixed in r1p1.

Description

Cortex-M3 asserts the SLEEPING output signal to indicate that Cortex-M3 is asleep and that HCLK may be stopped. In Cortex-M3 an instruction fetch's data-phase may overlap the assertion of SLEEPING. This overlap period may be extended by the HREADY associated with the instruction fetch being low.

Conditions

1. A WFI or WFE instruction is executed by Cortex-M3.
2. The core fetches another instruction.
3. The SLEEPING and data-phase of the instruction fetch overlap.

Implications

Ordinarily this will have no impact on Cortex-M3 provided that the AHB protocol is followed. Problems may arise if the bus infrastructure is using a different clock to Cortex-M3, for example, the HCLK to Cortex-M3 is gated but the clock to the bus is not. This would result in the data phase of the instruction fetch being completed by the bus but Cortex-M3 missing it due to not receiving the matching clock. This would result in Cortex-M3 potentially executing the wrong instruction when it woke from sleep.

Workaround

No workaround is required if the bus infra-structure uses the same clock. If the bus is using a different clock to Cortex-M3 then ensure that the clock to Cortex-M3 is not gated during the outstanding instruction fetch data-phase. This can be done by registering the address phase of the instruction fetch and holding it until the corresponding HREADY is asserted. After this period the clock to Cortex-M3 can be gated.

404812: DBGPWRUP generated incorrectly

Status

Affects:	product Cortex-M3, Cortex-M3 with ETM.
Fault status:	Impl, Present in: r1p0, Fixed in r1p1.

Description

The DBGPWRUP signal as part of the integration kit is generated incorrectly and causes some additional power to be used.

Conditions

1. Clock gating is implemented in Cortex-M3 Integration

Implications

Since DAPEN is permanently tied high it means that the clock gate for DAPCLK is therefore permanently enabled. This means that the DAPCLK will not be gated as much as it should be and therefore more power would be used than expected. The affects of DAPSELSWJ being used instead of CDBGPWRUPSWJ are masked due to DAPEN being tied high.

Workaround

This is a non-functional errata but it would cause more power to be used than expected.

Since these signals are in the CortexM3Integration file the implementer may alter the DBGPWRUP generation to not use the DAPEN and to use CDBGPWRUPSWJ instead of DAPSELSWJ.

To implement a fix change the following line in CortexM3Integration.v

from:

assign DBGPWRUP = DAPEN | DAPSELSWJ | CDBGPWRUPSW;

To:

assign DBGPWRUP = CDBGPWRUPSWJ | CDBGPWRUPSW;

531064: SWJ-DP missing POR reset sync

Status

Affects: product Cortex-M3, Cortex-M3 with ETM.

Fault status: Impl, Present in: r1p1,r1p1-00rel0,r1p1-01rel0, Fixed in r2p0-00rel0.

Description

In Cortex-M3 r0p0 and r1p0, the SWJ-DP has an internal reset synchronizer for the power on reset signal. Version r1p1 was upgraded to a newer version of SWJ-DP and this SWJ-DP did not have the same reset synchronizer inside. As a result of that the timing of the power on reset could potentially be a problem.

Implications

none

Workaround

The r1p1 CortexM3Integration level can be modified to add the reset synchronizer.

The signal which has the problem is the nPOTRST input on SWJ-DP. Add a few new signals:

```
 reg   nPOTRSTQ;  // DFF #1
 reg   nPOTRSTQQ; // DFF #2
 wire  inPOTRST;  // reset bypass MUX
// Add the synchroniser DFFs:
 // nPOTRST synchroniser
 always @ (posedge SWCLKTCK or negedge PORESETn)
   if (!PORESETn) begin
      nPOTRSTQ  <= 1'b0;
      nPOTRSTQQ <= 1'b0;
    end
   else begin
      nPOTRSTQ  <= 1'b1;
      nPOTRSTQQ <= nPOTRSTQ;
    end
// And a reset bypass MUX:
 assign inPOTRST   = RSTBYPASS ? PORESETn : nPOTRSTQQ;
// And connect the MUX output inPOTRST  to the nPOTRST input of SWJ-DP:
 DAPSWJDP uDAPSWJDP
    (// Inputs
     .nPOTRST            (inPOTRST),
```

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265