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 r0p0 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

20 Mar 2006: Changes in Document v2

Page	Status	ID	Cat	Summary
11	New	372726	Cat 2	Exit reset and exit debug causes additional E atom to be traced
12	New	375889	Cat 2	De-assertion of NIDEN might cause incorrect trace
13	New	377489	Cat 2	SysTick COUNTFLAG does not get cleared when the SysTick Current Value Register is written.
14	New	377490	Cat 2	SysTick COUNTFLAG automatic clear operation has priority over set.
15	New	377492	Cat 2	Incorrect return PC may be stacked for NMI in parallel with double fault lock up case
16	New	377493	Cat 2	C_MASKINTS in parallel with disabled interrupts can cause local faults to not be taken.
17	New	377494	Cat 2	Instruction incorrectly pre-fetch aborts due to miss-predicted branch
18	New	377496	Cat 2	Non byte size bit-band accesses in BE8 mode access incorrect bits
19	New	377497	Cat 2	Setting C_MASKINTS in the same cycle as halting debug is cleared
20	New	377519	Cat 2	HPROT always reports non-bufferable during HW stacking
21	New	377521	Cat 2	Interrupted alignment-faulting store following failing STREX may corrupt stack.
22	New	377522	Cat 2	SysTick CLKSOURCE is not forced to 1 when SysTick NOREF is high
23	New	377523	Cat 2	Fault on first of a pair of pipelined loads misinforms ETM interface
24	New	377524	Cat 2	Incorrect ETM Peripheral ID
25	New	377525	Cat 2	Authentication Status register reads as 0.
26	New	377681	Cat 2	Interrupted fault-generating load/store pair with SP base-writeback may corrupt stack
28	New	372716	Cat 3	RFE traced after Debug exit
29	New	377491	Cat 3	DWTTRAP/EXTERNAL bits in DFSR can be incorrectly set
31	New	377498	Cat 3	SVC/UNDEF swapped with branch inside HF/NMI lockup
30	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
9	New	369016	Cat 2	The second or third part of an unaligned transaction may be marked incorrectly
27	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
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
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
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

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, Open. Unchanged in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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.

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, Open. Updated in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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, Open. New in this document.

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.