ARM Processor Cortex®-A73 MPCore

Product Revision r0

Software Developers Errata Notice

Non-Confidential - Released

Software Developers Errata Notice

Non-Confidential Proprietary Notice

Web Address

http://www.arm.com

Feedback on content

If you have any comments on content, then send an e-mail to errata@arm.com . Give:

- the document title
- the document number, ARM-EPM-086451
- the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Release Information

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text in Chapter 2. Fixed errata are not shown as updated unless the erratum text has changed. The summary table in section 2.2 identifies errata that have been fixed in each product revision.

04 Oct 2016: Changes in Document v10

Page	Status	ID	Cat	Rare	Summary of Erratum
15	New	858072	CatB		A Cacheable Store-Release followed by a Non-cacheable Load-Acquire might not be observed in program order

04 Jul 2016: Changes in Document v9

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

08 Jun 2016: Changes in Document v8

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

08 Mar 2016: Changes in Document v7

Page	Status	ID	Cat	Rare	Summary of Erratum
14	New	855423	CatB		Broadcast maintainance operations might not be correctly synchronized between cores
29	New	855227	CatC		Access rights to register EDPRCR.CORENPDRQ is not fully compliant with ARM ARM

26 Jan 2016: Changes in Document v6

Page	Status	ID	Cat	Rare	Summary of Erratum
11	New	854222	CatA	Rare	A conjunction of several stores allocating in same L1 cache location might cause a deadlock
28	New	854223	CatC		Consecutive instructions that write partially flags might lead to {C,V} flags corruption

09 Dec 2015: Changes in Document v5

Page	Status	ID	Cat	Rare	Summary of Erratum
27	New	853676	CatC		Store-exclusives within a cluster might erroneously succeed

19 Oct 2015: Changes in Document v4

Page	Status	ID	Cat	Rare	Summary of Erratum
20	New	852425	CatC		Accessing Non implemented DBGBVR, DBGBCR, DBGWVR, DBGWCR can be trapped to Debug state instead of UNDEF exception
25	New	852921	CatC		An instruction that should be stepped by the software step (Resp. halting step) might not be executed
26	New	852971	CatC		EDSCR.A does not correctly flag pending physical and virtual system error interrupts

16 Sep 2015: Changes in Document v3

Page	Status	ID	Cat	Rare	Summary of Erratum
11	New	852428	CatA		Execution of instructions split into several micro-ops might cause deadlock
13	New	852427	CatB		Execution of 2 instructions with opposite condition code might lead to either a data corruption or a CPU deadlock
19	New	852424	CatC		Software step exception syndrome can wrongly show a LDX has been stepped
21	New	852426	CatC		Direct branch instructions executed before a trace flush might be output in an atom packet after flush acknowledgement
22	New	852471	CatC		ETM might trace an incorrect exception address
23	New	852472	CatC		ETM might lose counter events while entering wfx mode
24	New	852530	CatC		Possible deadlock when the ETM has overflowed and the core executes a WFE instruction

30 Jul 2015: Changes in Document v2

Page	Status	ID	Cat	Rare	Summary of Erratum
15	New	851571	CatC		External Debug PC Sample Register bit [0] might erroneously be set
17	New	851621	CatC		In AArch64, Crypto and SIMD instructions are counted as VFP in PMU events
18	New	851673	CatC		Wrong ESR.EC for a coprocessor access to the GIC interface hypervisor control registers generating undefined exception

17 Jun 2015: Changes in Document v1

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

Contents

Chapter 1.
Introduction

This chapter introduces the errata notice for the Cortex-A73 MPCore™ ARM Cortex-R5 and Cortex-R5F processors.

1.1. Scope of this document

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

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

This document describes errata that may impact anyone who is developing software that will run on implementations of this ARM product.

1.2. Categorization of errata

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

Table 1 Categorization of errata

Errata Type	Definition
Category A	A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.
Category A(rare)	A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category B	A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.
Category B(rare)	A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category C	A minor error.

Chapter 2.
Errata Descriptions

2.1. Product Revision Status

The r*n*p*n* identifier indicates the revision status of the product described in this book, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

2.2. Revisions Affected

Table 2 below lists the product revisions affected by each erratum. A cell marked with **X** indicates that the erratum affects the revision shown at the top of that column.

This document includes errata that affect revision r0 only.

Refer to the reference material supplied with your product to identify the revision of the IP.

Table 2 Revisions Affected

ID	Cat	Rare	Summary of Erratum	r0p0	r0p1	r0p2
852428	**CatA**		**Execution of instructions split into several micro-ops might cause deadlock**	**X**		
854222	**CatA**	**Rare**	**A conjunction of several stores allocating in same L1 cache location might cause a deadlock**	**X**	**X**	
858072	**CatB**		**A Cacheable Store-Release followed by a Non-cacheable Load-Acquire might not be observed in program order**	**X**	**X**	**X**
855423	**CatB**		**Broadcast maintainance operations might not be correctly synchronized between cores**	**X**	**X**	
852427	**CatB**		**Execution of 2 instructions with opposite condition code might lead to either a data corruption or a CPU deadlock**	**X**		
855227	**CatC**		**Access rights to register EDPRCR.CORENPDRQ is not fully compliant with ARM ARM**	**X**	**X**	
854223	**CatC**		**Consecutive instructions that write partially flags might lead to {C,V} flags corruption**	**X**	**X**	
853676	**CatC**		**Store-exclusives within a cluster might erroneously succeed**	**X**		
852971	**CatC**		**EDSCR.A does not correctly flag pending physical and virtual system error interrupts**	**X**		
852921	**CatC**		**An instruction that should be stepped by the software step (Resp. halting step) might not be executed**	**X**		
852530	**CatC**		**Possible deadlock when the ETM has overflowed and the core executes a WFE instruction**	**X**		
852472	**CatC**		**ETM might lose counter events while entering wfx mode**	**X**	**X**	**X**

ID	Cat	Rare	Summary of Erratum	r0p0	r0p1	r0p2
852471	CatC		ETM might trace an incorrect exception address	X		
852426	CatC		Direct branch instructions executed before a trace flush might be output in an atom packet after flush acknowledgement	X	X	X
852425	CatC		Accessing Non implemented DBGBVR, DBGBCR, DBGWVR, DBGWCR can be trapped to Debug state instead of UNDEF exception	X		
852424	CatC		Software step exception syndrome can wrongly show a LDX has been stepped	X		
851673	CatC		Wrong ESR.EC for a coprocessor access to the GIC interface hypervisor control registers generating undefined exception	X		
851621	CatC		In AArch64, Crypto and SIMD instructions are counted as VFP in PMU events	X		
851571	CatC		External Debug PC Sample Register bit [0] might erroneously be set	X		

2.2.1. r0p0 implementation fixes

The following errata might be fixed in some implementations of r0p0. This can be determined by reading the REVIDR register where a set bit indicates that the erratum is fixed in this part.

REVIDR[0]	852428 CatA Execution of instructions split into several micro-ops might cause deadlock **AND** 852427 CatB Execution of 2 instructions with opposite condition code might lead to either a data corruption or a CPU deadlock

Note that there is no change to the MIDR which remains at r0p0 but the REVIDR is updated as above to indicate which errata are corrected. Software will identify this release through the combination of MIDR/MIDR_EL1 and REVIDR/REVIDR_EL1.

2.2.2. r0p1 implementation fixes

The following errata might be fixed in some implementations of r0p1. This can be determined by reading the REVIDR register where a set bit indicates that the erratum is fixed in this part.

REVIDR[0]	854222 CatA A conjunction of several stores allocating in same L1 cache location might cause a deadlock

Note that there is no change to the MIDR which remains at r0p1 but the REVIDR is updated as above to indicate which errata are corrected. Software will identify this release through the combination of MIDR/MIDR_EL1 and REVIDR/REVIDR_EL1.

2.3. Category A

852428: Execution of instructions split into several micro-ops might cause deadlock

Category A

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

In AArch32 state, under very rare timing conditions, the CPU might deadlock upon execution of an conditional instruction split into several micro-ops and making use of "temporary registers", for which the condition code fails.

Configuration affected

This defect affects all configurations of the processor.

Conditions

The CPU must be in AArch32 and execute an instruction having the following characteristics:

* Either 1) In the A32 instruction set, the instruction is an arithmetical register-shifted register instruction, for example {{ADD r0, r1, r2, LSL r3}}, or a defined LDC instruction, for example {{LDC p14 to DBGDTRTXint}}.

Or 2) In the A32 or T32 instruction sets, the instruction is one of {{SXTAB16}}, {{SXTAB}}, {{SXTAH}}, {{UXTAB16}}, {{UXTAB}}, {{UXTAH}}.

* And the instruction is conditional

* And its condition code fails.

Note that other timing conditions are required to trigger the erratum.

Implications

In AArch32, under rare timing conditions, a deadlock might occur.

Workaround

There is no workaround

2.4. Category A (Rare)

854222: A conjunction of several stores allocating in same L1 cache location might cause a deadlock

Category A Rare

Products Affected: Cortex-A73 MPCore.

Present in: r0p0, r0p1

Description

Under very rare timing conditions, a conjunction of several stores occurring at same L1 cache location might initiate a coherent transaction not linked to any of the pending stores, which leads to a deadlock.

Configuration affected

This erratum affects all configurations of the processor.

Conditions

The following conditions are necessary for this to occur:

* A store on cache line A enters the STB and gets the old way information from a previous and completed store in same cache line. This new store takes a lot of time to do the lookup in the cache. Meanwhile:

* A new cache line B is allocated at same index/way hold by the store on line A and the cache line A is also allocated in Shared state in the cache by a load or a prefetch.

* A new cache line C is then allocated in place of line B. Cache line B will therefore be evicted.

* The STB finally does the lookup for line A and hits in Shared at the same time the eviction of line B is started, causing an unexpected eviction hazard.

* The hit Shared information initiates a coherent transaction to the SCU but the eviction hazard forces the STB to replay the lookup and therefore does not expect to wait for this coherent transaction.

* The coherent transaction response is received by the core and causes a deadlock because it is not linked to a pending request.

Note that other timing conditions and further micro-architecture events are required to trigger this erratum.

Implications

Under rare timing conditions, a deadlock might occur.

Workaround

There is no workaround for this erratum.

2.5. Category B

852427: Execution of 2 instructions with opposite condition code might lead to either a data corruption or a CPU deadlock

Category B

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

In AArch32 state, under very rare timing conditions, upon execution of a sequence of two instructions having opposed condition codes, the CPU might deadlock or corrupt the value of the destination register.

Configuration affected

This defect affects all configurations of the processor.

Conditions

The cpu must be in AArch32 state and executes two instructions, back to back in execution order, with the following characteristics:

* the two instructions must be conditional, with resolution of their condition codes being opposed;

* each of the two instructions must have either only one operand, or two operands in which case the destination register must be identical to one of the two operand registers.

Ex: MOVEQ r0, r1 followed by ADDNE r0, r0, r2

Note that other timing conditions are required to trigger the erratum.

Implications

Under rare timing conditions, a data corruption or a deadlock might occur.

Workaround

Setting bit [12] of the Diagnostic Register prevents this erratum from occurring:

```
MRC p15, 0, r0, c15, c0, 1
ORR r0, r0, #1<<12
MCR p15, 0, r0, c15, c0, 1
```

855423: Broadcast maintainance operations might not be correctly synchronized between cores

Category B

Products Affected: Cortex-A73 MPCore.

Present in: r0p0, r0p1

Description

Execution of a DSB instruction should synchronize all broadcast cache and branch predictor maintainance operations between all masters in the same coherency domain. Due to this erratum, a core executing a DSB instruction while under low utilization might fail to update other masters.

Configuration affected

This erratum affects all configurations of the processor

Conditions

1. The core has no pending loads, stores or system instructions in any stage of its pipeline.

2. Due to rare circumstances, the memory system is completely empty.

3. The core executes an instruction cache or a branch prediction maintenance operation (DVM) that is broadcasted to the other cores in the same coherency domain.

4. The DVM operation is not properly detected by the system.

5. A subsequent DSB fails to ensure that the DVM is visible.

Implications

This erratum might lead to data corruption

Workaround

Setting bit[7] of the Power Diagnostic register prevents this erratum from occurring:

In AArch32:

MRC p15, 0, r0, c15, c0, 2

ORR r0, r0, #1<<7

MCR p15, 0, r0, c15, c0, 2

In AArch64:

MRS r0, S3_0_C15_C0_2

ORR r0, r0, #1<<7

MSR S3_0_C15_C0_2, r0

Setting this bit has a very negligible impact on power.

858072: A Cacheable Store-Release followed by a Non-cacheable Load-Acquire might not be observed in program order

Category B

Products Affected: Cortex-A73 MPCore.

Present in: r0p0, r0p1, r0p2

Description

A STRL/STLXR Cacheable instruction, followed by an LDAR Non-cacheable instruction, might not follow the program order. Because of this erratum, the LDAR instruction could be made visible before the STRL/STLXR instruction whereas it should not, as specified in the ARMv8 architecture.

Configurations Affected

This erratum affects all configurations of the processor.

Conditions

The CPU executes a code containing a Store-Release instruction targeting normal Cacheable memory and a subsequent Load-Acquire instruction targeting Normal Non-cacheable memory or Device memory.

The Store-Release instruction can be exclusive (STLXR), but the Load-Acquire instruction cannot.

Under rare circumstances, the LDAR instruction might be executed before the STRL/STLXR instruction has been observed by the other CPUs.

This breaks the Load-Acquire/Store-Release ordering requirement defined in the ARMv8 architecture.

Implications

A program that mixes Cacheable and Non-cacheable Store-Release/Load-Acquire instructions and relies on their ordering could end up with data corruption.

Workaround

A DMB between the Store-Release instruction and the Load-Acquire instruction fixes this erratum.

2.6. Category B (Rare)

There are no errata in this category

2.7. Category C

851571: External Debug PC Sample Register bit [0] might erroneously be set

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

EDPCSR bit [0] should be zero when the EDPCSR holds the address of a recently executed instruction. Because of this erratum, if the instruction address corresponds to an instruction executed in T32 state, then the least-significant bit of the EDPCSR may erroneously be set to one.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

1. The processor executes an instruction in T32 state.

2. The address of this instruction is sampled into the EDPCSR.

3. The EDPCSR is read.

Implications

The EDPCSR may have PC sample values with the least significant bit set.

Workaround

The least significant bit of the EDPSCR should be ignored when it contains an instruction address.

851621: In AArch64, Crypto and SIMD instructions are counted as VFP in PMU events

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

Speculative execution of Advanced SIMD and Cryptographic Extension instructions while in AArch64 state should be measurable via the ASE_SPEC and CRYPTO_SPEC PMU events. Because of this erratum, events that should update ASE_SPEC or CRYPTO_SPEC erroneously increment VFP_SPEC instead.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

1. A PMU counter is configured to count ASE_SPEC, CRYPTO_SPEC, or VFP_SPEC.

2. An AArch64 Advanced SIMD or Cryptographic Extension instruction is speculatively executed.

Implications

ASE_SPEC, CRYPTO_SPEC, and VFP_SPEC events are indistinguishable, and are all accumulated by VFP_SPEC while in AArch64. This erratum does not impact performance monitoring in AArch32 state.

Workaround

There is no workaround for this erratum.

851673: Wrong ESR.EC for a coprocessor access to the GIC interface hypervisor control registers generating undefined exception

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

In EL0 AArch32 state, on an access to a GIC CPU interface hypervisor control register, the ESR.EC reported will be 0x3 instead of 0x0.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

The issue can be found on the following cases:

Any access from EL0 AArch32 to an ICH_* coprocessor register, in secure state or with HCR_EL2.TGE=0. EL1 AArch64 processes the exception and ESR_EL1.EC is wrong.

Any access from EL0 AArch32 to an ICH_* coprocessor register, in non-secure state with HCR.TGE=1 or HCR_EL2.TGE=1. EL2 processes the exception and either HSR.EC or ESR_EL2.EC is wrong.

Implications

This impacts any software running in EL0 AArch32 state, supporting hypervisor, and using the GIC interface (GICv3 and above).

Workaround

There is no workaround for this erratum.

852424: Software step exception syndrome can wrongly show a LDX has been stepped

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

When a software step (Resp. halting step) is set on an instruction followed by a Load-exclusive class instruction, the corresponding software step exception (Resp. halting step debug event) will set ESR.ISS[6] (Resp. will set EDSCR.STATUS=6'b011111), incorrectly indicating that the stepped instruction is a Load-exclusive class instruction.

Note that ESR.ISS[6] (Resp. EDSCR.STATUS) is set correctly if the stepped instruction is really a Load-exclusive instruction.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

The erratum might occur if the code includes Load-exclusive class instructions and the software or halting step is enabled.

Implications

The software running on the target (Resp. the debugger software) will execute some maintenance code for semaphore take and release which is not necessary.

Workaround

If ESR.ISS[6] is set (Resp. EDSCR.STATUS=6'b011111), the software step handler (Resp. the debugger maintenance) should check if the stepped instruction is really a Load-exclusive instruction before running the related specific code.

852425: Accessing Non implemented DBGBVR, DBGBCR, DBGWVR, DBGWCR can be trapped to Debug state instead of UNDEF exception

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

In AArch64 state, when EDSCR.TDA is set and the conditions are correct to cause a Software Access debug event (trap to debug state), accessing non implemented DBGBVR, DBGBCR, DBGWVR, DBGWCR, respectively for breakpoints 6 to 15 and for watchpoint 4 to 15 will generate a trap to debug state instead of an UNDEF exception.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

When the core is in AArch64 state and the following conditions exist:

EDSCR.TDA is set

Halting is allowed

OSLK is cleared.

Any access to DBGBVR(6-15), DBGBCR(6-15), DBGWVR(4-15), DBGWCR(4-15) made through MRS, MSR instructions will trigger the erratum.

Implications

The debugger gets the control over the CPU by entering the debug state at a time when it is not expected to do so.

DLR is not correctly updated on the debug state entry so that the debugger cannot exit properly by returning to the correct PC.

Workaround

Software should only rely on implemented DBGBVR, DBGBCR, DBGWVR, DBGWCR registers to trap them to debug state.

There is no specific workaround.

This bug can be detected by checking EDSCR.STATUS is set at "Software access to debug register" value and after executing an UNDEF instruction through the ITR, the corresponding ELR is set to the UNDEF handler address.

852426: Direct branch instructions executed before a trace flush might be output in an atom packet after flush acknowledgement

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0, r0p1, r0p2

Description

The Embedded Trace Macrocell (ETMv4) architecture requires that when a trace flush is requested on the AMBA Trace Bus (ATB), a processor must complete any packets that are in the process of being encoded. A processor must output these packets before acknowledging the flush request. When trace is enabled, the processor attempts to combine multiple direct branch instructions into a single Atom packet. If a direct branch instruction is executed and an Atom packet is in the process of being generated, the processor does not force completion of the packet before acknowledging the flush request. This is a violation of the ETMv4 architecture.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

1. ETM is enabled.

2. Instruction tracing is active.

3. One or more direct branch instructions are executed.

4. An Atom packet is being encoded but is not complete.

5. A trace flush is requested on the AMBA ATB.

Implications

When the above conditions occur, the Atom packet being encoded must complete and be output before the trace flush request being acknowledged. Because of this erratum, the Atom packet is output after the flush is acknowledged. Therefore, it will appear to the software monitoring the trace that the direct branch was executed after the requested flush.

Workaround

Enabling the timestamp by setting TRCCONFIGR.TS solves the issue because it completes the atom packets through the timestamp behavior.

852471: ETM might trace an incorrect exception address

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

The address in an exception packet should be the address of the interrupted instruction. Due to this erratum, the address might be equal to the first address of the exception. The trace stream is not corrupted, and decompression can continue after the affected packet.

Configurations Affected

This erratum affects all configurations of all the processors.

Conditions

The following sequence is required to hit this erratum:

1. The ETM must start tracing instructions because of one of the following:

 * Viewinst goes high.

 * The security state changes such that trace is now permitted.

 * The values of the external debug interface (DBGEN, SPIDEN, NIDEN, SPNIDEN) change such that trace is now permitted.

 * The core exits debug mode.

2. Before the core executes any other behavior which would cause trace to be generated, it executes a direct branch instruction, which might be taken or not-taken.

3. The next instruction is a load or store that takes a Data Abort or Watchpoint exception.

After this sequence, provided certain timing specific conditions are met, the address in the exception packet might be incorrect.

Implications

The trace decompressor might incorrectly infer execution of many instructions from the branch target to the provided address.

Workaround

The trace decompressor can detect that this erratum has occurred by checking if the exception address is in the Vector Table and the branch was not expected to be taken to the Vector Table.

A decompressor can infer the correct address of the exception packet. It will be given by the target of the preceding branch (If the branch was taken), or the next instruction after the branch (If the branch was not-taken).

852472: ETM might lose counter events while entering wfx mode

Category C
Products Affected: Cortex-A73 MPCore.
Present in: r0p0, r0p1, r0p2

Description

If the ETM resources become inactive because of low-power state, there is a one-cycle window during which the counters and the sequencer might ignore counter-at-zero resources.

Configurations Affected

This erratum affects all configurations of the processor.

Conditions

The following sequence is required to hit this erratum:

1. The core executes a WFI or WFE instruction.

2. The ETM enters low-power state because of this.

3. In a one-cycle window around this point, either:

   * A counter in self-reload mode generates a counter-at-zero resource.

   * A counter in normal mode gets a RLDEVENT on the cycle in which it has just transitioned to zero.

4. A counter or sequencer is sensitive to the counter-at-zero resource.

Implications

Counters sensitive to a counter-at-zero resource might not reload or decrement. If the sequencer is sensitive to a counter-at-zero resource, it might not change state, or might change to an incorrect state.

Workaround

The ETM can be prevented from entering low-power mode by programming LPOVERRIDE bit of TRCEVENTCTL1R to 1. This workaround is only needed if there is a counter or sequencer sensitive to a counter-at-zero resource, and is not normally necessary.

852530: Possible deadlock when the ETM has overflowed and the core executes a WFE instruction

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

After an overflow of the ETM exit fifo, The ETM could miss the overflow exit state if the core interface fifo overflows.

Due to this second overflow, the ETM might never exit from an overflow state and never respond to a flush request such as a WFI/WFE entry and deadlock the core on WFE and STANDBYWFI will be never set on WFI.

Configurations Affected

This erratum affects all configurations of all the processors.

Conditions

The following sequence is required to hit this erratum:

1. The ETM must allow overflow of the Core interface with the TRCAUXCTLR.INOVFLOWEN set,

2. The trace output fifo has overflowed and is empty,

3. At exactly the same cycle, the core interface fifo overflows.

After this sequence, provided certain timing specific conditions are met, the ETM will be stuck in an Overflow state.

Implications

Under rare timing conditions, a deadlock of the ETM and the Core might occur.

Workaround

The preferred workaround is:

* Unset TRCAUXCTLR.INOVFLOWEN to prevent any overflow of the Core interface fifo.

In case the preferred workaround is not possible:

* Set TRCAUXCTLR.IDLEACKOVERRIDE to force the ETM idle acknowledgement, thus preventing a deadlock of the core on a WFE entry.

852921: An instruction that should be stepped by the software step (Resp. halting step) might not be executed

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

If the stepped instruction is predicted as a branch and is not actually a branch instruction, this instruction will not be executed. However, the software step exception (Resp. debug entry) will be executed.

Configuration affected

This erratum affects all configurations of the processor.

Conditions

The erratum might be triggered upon execution of an instruction having the following characteristics:

* The processor should step an instruction with the software (Resp. halting) step state machine.

* The stepped instruction is not a branch.

* The stepped instruction is predicted as a branch by the branch prediction unit. This means a branch at the same virtual address as the stepped instruction has previously been executed by the processor.

Implications

The software (Resp. halting) step state machine will not execute the stepped instruction.

Workaround

There is no workaround for this erratum.

However, the debugger can detect this defect in the single step debug handler by comparing the current value of DLR with the former one from the previous stepped instruction.

852971: EDSCR.A does not correctly flag pending physical and virtual system error interrupts

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

When in Debug state, EDSCR.A does not flag correctly pending physical and virtual SError interrupts (SEI, REI, and VSEI). It is set instead on other pending sources of interrupts which can be asynchronous data aborts, physical and virtual IRQ and FIQ, and virtual SError interrupts from HCR.VA/HCR_EL2.VSE.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

The processor is in Debug state with the given interrupt sources enabled.

Implications

EDSCR.A might read as 1 when there is no physical or virtual SError interrupt pending, meaning the debugger will execute some maintenance code which is not relevant given that the pending interrupts are not the expected one and might have been cleared. EDSCR.A might read as 0 when an SError interrupt is pending.

Workaround

The debugger must not rely on EDSCR.A. The debugger should change to EL1 or above and read ISR_EL1 through the ITR.

853676: Store-exclusives within a cluster might erroneously succeed

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0

Description

The architecture requires that a Store-Exclusive should fail if a second processor performs a Store-Exclusive to the same location after the original processor performed its Load-Exclusive.

Due to this erratum, under unusual cluster configurations combined with very rare timing circumstances, a continuous stream of back-to-back stores that are within the same 64-bits as the address being acted upon by the Store-Exclusive can result in the original processor?s Store-Exclusive erroneously succeeding even though the second processor has already updated the location.

This erratum only impacts interactions between processors within a single cluster.

Configurations Affected

This erratum affects all configurations of the processor.

Conditions

The following sequence is required to happen for this erratum to occur:

1. A core must be running at least 5 times slower than another one in the cluster.

2. The semaphore must be a subset of 64-bit data, and the remaining bytes are accessed by the core running at the slower frequency.

3. A STR releases this semaphore, followed by a stream of stores merging into the same 64-bit data.

4. Meanwhile during this uninterrupted stream, another core (running at least 5 times faster) executes a LDREX/STREX sequence and the STREX succeeds.

5. Immediately after the stream of stores, the `slow? core executes a LDREX/STREX sequence and under rare timing conditions, the STREX also succeeds although it should not.

6. Finally, 2 cores have locked the semaphore, which might lead to data corruption.

Implications

2 cores might have access to shared data protected by semaphore, which would cause data corruption.

Workaround

There is no workaround for this erratum.

854223: Consecutive instructions that write partially flags might lead to {C,V} flags corruption

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0, r0p1

Description

In AArch32 state, under very specific code sequence and very rare timing conditions, the flags C and V might be corrupted.

Configurations Affected

This erratum affects all configurations of the processor

Conditions

The following sequence is required to happen for this erratum to occur:

* In AArch32 state, an instruction produces NZCV flags, followed by a sequence including 19 instructions that all produce only a subset of these flags

* The first instruction of this sequence has still not produced its flags while the last instruction is stalled before being dispatched.

* This sequence is followed by a stalled instruction that needs to read C and/or V flags, produced by the first instruction.

* Finally, a VMRS/FMSTAT producing all flags NZCV is then executed at the exact moment when the first instruction of the sequence completes

* Under further very rare timing conditions, C and/or V flags produced by VMRS/FMSTAT might corrupt the ones from the first instruction.

Implications

The flags C and/or V flags might be corrupted on a code sequence that cannot be generated by any compiler

Workaround

There is no workaround to this erratum.

855227: Access rights to register EDPRCR.CORENPDRQ is not fully compliant with ARM ARM

Category C

Products Affected: Cortex-A73 MPCore.

Present in: r0p0, r0p1

Description

In certain conditions, the processor incorrectly ignores writes to EDPRCR.CORENPDRQ from the debugger.

Configurations Affected

This erratum affects all configurations of the processor.

Conditions

1. Either:

   DBGEN == 0

   DBGEN == 1, SPIDEN == 0, and either SDCR.EDAD == 1 or MDCR_EL3.EDAD == 1.

2. The debugger writes to the EDPRCR register.

In these conditions, the processor incorrectly ignores the value written to EDPRCR[0], meaning that EDPRCR.CORENPDRQ is unchanged.

Implications

Because the write is ignored, the DBGNOPWRDWN pin might not be asserted as expected. As a consequence, the system might not emulate power-down of the processor core power domain and instead remove power, meaning that the state of the debug session is lost.

However, note that because these conditions arise when the system or software is otherwise restricting a debugger's capabilities, the impact of removing power might be reduced. In particular, if DBGEN is tied LOW, most of the state lost on power down could not have been used by the debugger.

Also note that direct writes to DBGPRCR_EL1.CORENPDRQ by software executing on the processor are unaffected.

Workaround

The workaround is for the external debugger to set EDPRCR.COREPURQ instead of EDPRCR.CORENPDRQ. This drives DBGPWRUPREQ, and the system must use both signals to emulate powerdown:

If DBGPWRUPREQ == 1 OR DBGNOPWRDWN == 1, the system should emulate power down of the core power domain, otherwise the system can power down the core power domain.