Cortex-A75 (MP058)
Software Developer Errata Notice

This document contains all known errata since the r0p0,r1p0,r1p1, r2p0, r2p1, r3p0, r3p1 release of the product.

Non-Confidential Proprietary notice

LES-PRE-20349

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Product Status

The information in this document is for a product in development and is not final.

Web address

http://www.arm.com/.

Feedback on this product

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

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on this document

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

- The document title.
- The document number: SDEN-859515.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Contents

Introduction

Scope

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

- The current status of the erratum.
- Where the implementation deviates from the specification and the conditions required for erroneous behavior to occur.
- The implications of the erratum with respect to typical applications.
- The application and limitations of a workaround where possible.

Categorization of errata

Errata are split into three levels of severity and further qualified as common or rare:

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.

Change control

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. Fixed errata are not shown as updated unless the erratum text has changed. The errata summary table on page 10 identifies errata that have been fixed in each product revision.

12-Mar-2020: Changes in document version 9.0				
ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

30-Jul-2018: Changes in document version 8.0				
ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

10-Mar-2018: Changes in document version 7.0				
ID	Status	Area	Cat	Summary of erratum
1058157	New	Programmer	CatC	Incorrect ETM timestamp value when timestamp and event generation happen on same cycle
1058155	New	Programmer	CatC	Cycle count value in timestamp packet might be incorrect

06-Nov-2017: Changes in document version 6.0				
ID	Status	Area	Cat	Summary of erratum
No new or updated errata in this document version.				

12-Jun-2017: Changes in document version 5.0				
ID	Status	Area	Cat	Summary of erratum
869960	New	Programmer	CatC	A PMU snapshot request returns erroneous value when PMU counters are disabled
858572	New	Programmer	CatC	Vector catch can be generated to EL1 AArch64
815950	New	Programmer	CatC	Using a 52-bit translation table base address does not report an Address Size fault
801389	New	Programmer	CatC	Accesses to some cluster Performance Monitor system registers are Read-As-Zero, Writes Ignored
836130	New	Programmer	CatC	Corrected Error Counter Repeat is incremented on the detection of any first error
820018	New	Programmer	CatC	Exceptions taken from Aarch32 to Aarch64 might generate an Address Size Fault when the corresponding VBAR_ELx is using a tagged address
900547	New	Programmer	CatC	ESB in Debug state does not defer SErrors

877067	New	Programmer	CatC	The dirty state bit of a translation table descriptor might erroneously be updated
872089	New	Programmer	CatC	SError taken as a result of stepping an ESB instruction might incorrectly set EDESR.SS
832595	New	Programmer	CatC	EDSCR.INTdis might erroneously cause asynchronous exceptions routed to EL2 to be masked
772669	New	Programmer	CatC	Single stepping might fail to step an instruction
771861	Updated	Programmer	CatB	Access to ATCR_EL1 is not trapped when HCR.TVM or HCR.TRVM are set

06-Feb-2017: Changes in document version 4.0				
ID	Status	Area	Cat	Summary of erratum
808979	Updated	Programmer	CatA	Half Precision Multiply and Accumulate might return a wrong result
792334	Updated	Programmer	CatC	PSTATE.UAO might affect LDR*T/STR*T instructions executed in AArch32
792226	Updated	Programmer	CatC	Leaving Debug state from EL0 does not restore PSTATE.PAN and PSTATE.UAO from DSPSR
782705	Updated	Programmer	CatC	Aborts on AT operations are ignored in Debug State
774033	Updated	Programmer	CatC	External changes to EDSCR.HDE might lead to unauthorized exception generation
769222	Updated	Programmer	CatC	Error not correctly recorded if reported when software is clearing ERR0STATUS
790861	Updated	Programmer	CatC	IESB might fail to synchronize a pending SError
790748	Updated	Programmer	CatB	Internal timing conditions might cause the CPU to stop processing interrupts
787961	Updated	Programmer	CatB	DRPS or Debug Exit with SCTLR_ELx.IESB set and a pending vSError causes erroneous behavior

12-Dec-2016: Changes in document version 3.0				
ID	Status	Area	Cat	Summary of erratum
770356	Updated	Programmer	CatC	Infinite loop of Store instructions might prevent older STR instruction from completing
784250	Updated	Programmer	CatB	A direct write to HCR_EL2.E2H might not be visible after execution of an ISB
782841	Updated	Programmer	CatC	Alignment fault on ERET might report an incorrect address tag in IFAR
783218	Updated	Programmer	CatC	DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might be erroneous

776926	Updated	Programmer	CatB	TLB entries of 1GB might not be invalidated
767916	Updated	Programmer	CatC	An exclusive sequence encountering poisoned data might not be able to progress
761663	Updated	Programmer	CatB	Wrong DFSR and IFSR descriptor format when taking an external abort from Non-secure to EL3 in AArch32
780153	Updated	Programmer	CatC	MSB of PMPCSR do not report sign extension of the Program Counter
754852	Updated	Programmer	CatC	EDPRCR.CORENPDRQ might not get set after a Cold reset
751005	Updated	Programmer	CatC	An MCR to ERXFR2 register in non-user mode does not UNDEF
739639	Updated	Programmer	CatB (rare)	Erroneous allocation of Page Table Entries might lead to a conflict abort
739130	Updated	Programmer	CatC	Inversion in Trap priority between ICH_HCR.TALL0/TALL1 and SCR.FIQ/IRQ
764081	Updated	Programmer	CatC	Implicit Error Synchronization Barrier might not be correctly generated

14-Nov-2016: Changes in document version 2.0				
ID	Status	Area	Cat	Summary of erratum
764571	Updated	Programmer	CatC	PC Sample register and Context ID registers are not consistent
771824	Updated	Programmer	CatC	DBGDSCRext.MOE, DBGDSCRint.MOE might get corrupted upon entering Debug state
772905	Updated	Programmer	CatB	Accesses to ETM registers trough the APB while the core is in Warm reset might hang
754402	Updated	Programmer	CatC	ESR, IFSR, and HSR might get corrupted when entering debug state
770802	Updated	Programmer	CatC	ETM might not generate an event packet and ATB trigger
770785	Updated	Programmer	CatC	ETM might assert AFREADY before all trace has been output

29-Sep-2016: Changes in document version 1.0				
ID	Status	Area	Cat	Summary of erratum
753282	Updated	Programmer	CatC	Executing an ESB in Software Step might cause a synchronized SError to be deferred instead of taken
750234	Updated	Programmer	CatC	ETM might trace an incorrect branch PC on taking an SError synchronized by an ESB

Errata summary table

The errata associated with this product affect product versions as below.

ID	Cat	Summary	Found in versions	Fixed in version
869960	CatC	A PMU snapshot request returns erroneous value when PMU counters are disabled	r1p0, r0p0, r1p1	r2p0
858572	CatC	Vector catch can be generated to EL1 AArch64	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
815950	CatC	Using a 52-bit translation table base address does not report an Address Size fault	r1p0, r0p0, r1p1	r2p0
801389	CatC	Accesses to some cluster Performance Monitor system registers are Read-As-Zero, Writes Ignored	r1p0, r1p1	r2p0
808979	CatA	Half Precision Multiply and Accumulate might return a wrong result	r1p0, r0p0	r1p1
770356	CatC	Infinite loop of Store instructions might prevent older STR instruction from completing	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
1058157	CatC	Incorrect ETM timestamp value when timestamp and event generation happen on same cycle	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
1058155	CatC	Cycle count value in timestamp packet might be incorrect	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
836130	CatC	Corrected Error Counter Repeat is incremented on the detection of any first error	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
820018	CatC	Exceptions taken from Aarch32 to Aarch64 might generate an Address Size Fault when the corresponding VBAR_ELx is using a tagged address	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
900547	CatC	ESB in Debug state does not defer SErrors	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
877067	CatC	The dirty state bit of a translation table descriptor might erroneously be updated	r1p0, r0p0, r1p1	r2p0
872089	CatC	SError taken as a result of stepping an ESB instruction might incorrectly set EDESR.SS	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
832595	CatC	EDSCR.INTdis might erroneously cause asynchronous exceptions routed to EL2 to be masked	r1p0, r0p0, r1p1, r2p0, r2p1, r3p0, r3p1	Open
792334	CatC	PSTATE.UAO might affect LDR*T/ STR*T instructions executed in AArch32	r0p0	r1p0
792226	CatC	Leaving Debug state from EL0 does not restore PSTATE.PAN and PSTATE.UAO from DSPSR	r0p0	r1p0
784250	CatB	A direct write to HCR_EL2.E2H might not be visible after execution of an ISB	r0p0	r1p0

ID	Cat	Summary	Found in versions	Fixed in version
782841	CatC	Alignment fault on ERET might report an incorrect address tag in IFAR	r0p0	r1p0
782705	CatC	Aborts on AT operations are ignored in Debug State	r0p0	r1p0
783218	CatC	DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might be erroneous	r0p0	r1p0
776926	CatB	TLB entries of 1GB might not be invalidated	r0p0	r1p0
764571	CatC	PC Sample register and Context ID registers are not consistent	r0p0	r1p0
767916	CatC	An exclusive sequence encountering poisoned data might not be able to progress	r0p0	r1p0
761663	CatB	Wrong DFSR and IFSR descriptor format when taking an external abort from Non-secure to EL3 in AArch32	r0p0	r1p0
780153	CatC	MSB of PMPCSR do not report sign extension of the Program Counter	r0p0	r1p0
771824	CatC	DBGDSCRext.MOE, DBGDSCRint.MOE might get corrupted upon entering Debug state	r0p0	r1p0
772669	CatC	Single stepping might fail to step an instruction	r0p0	r1p0
771861	CatB	Access to ATCR_EL1 is not trapped when HCR.TVM or HCR.TRVM are set	r0p0	r1p0
772905	CatB	Accesses to ETM registers trough the APB while the core is in Warm reset might hang	r0p0	r1p0
774033	CatC	External changes to EDSCR.HDE might lead to unauthorized exception generation	r0p0	r1p0
769222	CatC	Error not correctly recorded if reported when software is clearing ERR0STATUS	r0p0	r1p0
790861	CatC	IESB might fail to synchronize a pending SError	r0p0	r1p0
790748	CatB	Internal timing conditions might cause the CPU to stop processing interrupts	r0p0	r1p0
787961	CatB	DRPS or Debug Exit with SCTLR_ELx.IESB set and a pending vSError causes erroneous behavior	r0p0	r1p0
754852	CatC	EDPRCR.CORENPDRQ might not get set after a Cold reset	r0p0	r1p0
754402	CatC	ESR, IFSR, and HSR might get corrupted when entering debug state	r0p0	r1p0
753282	CatC	Executing an ESB in Software Step might cause a synchronized SError to be deferred instead of taken	r0p0	r1p0

ID	Cat	Summary	Found in versions	Fixed in version
751005	CatC	An MCR to ERXFR2 register in non-user mode does not UNDEF	r0p0	r1p0
750234	CatC	ETM might trace an incorrect branch PC on taking an SError synchronized by an ESB	r0p0	r1p0
739639	CatB (rare)	Erroneous allocation of Page Table Entries might lead to a conflict abort	r0p0	r1p0
739130	CatC	Inversion in Trap priority between ICH_HCR.TALL0/TALL1 and SCR.FIQ/IRQ	r0p0	r1p0
770802	CatC	ETM might not generate an event packet and ATB trigger	r0p0	r1p0
770785	CatC	ETM might assert AFREADY before all trace has been output	r0p0	r1p0
764081	CatC	Implicit Error Synchronization Barrier might not be correctly generated	r0p0	r1p0

Errata descriptions

Category A

808979

Half-precision Multiply and Accumulate might return a wrong result

Status

Affects: Cortex-A75
Fault Type: Programmer Category A
Fault Status: Present in r0p0 and r1p0.

Description

With a very limited set of normalized operands, a half-precision (FP16) Fused-Multiply-Add type and Fused-Multiply-Subtract type instruction can return either:

- A minimum normal result (0x0400 or 0x8400) or a rounded-up value (0x0401 or 0x8401) instead of a zero result, usually when the FZ16 bit is not set.
- A maximum result (0x7BFF or 0xFBFF) or an infinity (0x7C00 or 0xFC00) instead of a zero result, depending on rounding mode, usually when the FZ16 bit is set.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions are required to trigger this erratum:

- The processor executes a half-precision (FP16) Fused-Multiply-Add type or Fused-Multiply-Subtract type instruction.
- The operands are such that the result should be zero.

Implications

The result reported is not zero.

Workaround

There is no workaround.

Category A (rare)

There are no errata in this category.

Category B

784250

A direct write to HCR_EL2.E2H might not be visible after execution of an ISB

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

A synchronization operation which does not change the translation regime might not ensure that indirect reads of HCR_EL2 observe the latest direct writes to HCR_EL2.E2H.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum occurs when the following conditions are met:

- The PE is at EL2 and system registers are visible for indirect reads.
- The PE executes a direct write to HCR_EL2, modifying the E2H field but not modifying the TGE field.
- Since the last synchronization operation, and until the next synchronization operation, no other changes are performed to system registers that affect the current translation regime.
- After the next synchronization operation, the PE uses the same translation regime.

Implications

If the PE has set HCR_EL2.E2H to 0, this erratum leaves TTBR1 as valid, which might lead to data corruption.
If the PE has set HCR_EL2.E2H to 1, this erratum prevents TTBR1 from becoming valid and will generate a spurious translation fault. On top of that, use of TTBR0 for translation might lead to data corruption.

Workaround

Changes to HCR_EL2.E2H must be done with another write to a system register that affects the current translation regime before going through a synchronization operation.
ARM does not expect changes to HCR_EL2.E2H to occur without further changes to other system registers that affect the current translation regime, for example a change to HCR_EL2.TGE.
Note that a write of any value to any other system register affecting the translation regime clears this issue.

776926

TLB entries of 1GB might not be invalidated

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

A TLBI IPAS2(L)E1(IS) invalidation in AArch64, or a TLBIIPAS2(L) invalidation in AArch32, might fail to invalidate intermediate caching of IPA to PA TLB entries, resulting in a wrong IPA to PA mapping being used for future translations.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

1. Stage 2 is enabled.
2. EL2 is running in AArch64 and using 4K granule, or EL2 is running in AArch32.
3. Stage 2 maps a 1GB block.
4. EL2 uses either Sequence 2 or Sequence 3 of the ARM ARM, as reproduced below, to invalidate a 1GB stage 2 translation.
5. This invalidation of a 1GB stage 2 translation is not followed by either a TLBIALL or a TLBIALLNSNH in AArch32, or a TLB VMALLE1 in AArch64.

Sequence 2:
The following code is sufficient to invalidate all cached copies of the stage 2 translations of the IPA held in Xt used to translate the virtual address VA (and the specified ASID when executing TLBI VAE1) held in Xt2, with the corresponding requirement for the broadcast versions of the instructions:

```
TLBI IPAS2E1, Xt
DSB
TLBI VAE1, Xt2
; or TLBI VAAE1, Xt2
```

Sequence 3:
The following code is sufficient to invalidate all cached copies of the stage 2 translations of the IPA held in Xt used to translate the IPA produced by the last level of stage 1 translation table lookup for the virtual address VA (and ASID when executing TLBI VALE1) held in Xt2, with the corresponding requirement for the broadcast versions of the instructions:

```
TLBI IPAS2E1, Xt
DSB
TLBI VALE1, Xt2
; or TLBI VAALE1, Xt2
```

Implications

This erratum might lead to data corruption, as a stale translation might remain cached in the Cortex-A75 TLB.

Workaround

A workaround consists in using the code defined as Sequence 1 in the ARM ARM for invalidating 1GB entries.

For example, in AArch64:

```
TLBI IPAS2E1, Xt
DSB
```

```
TLBI VMALLE1
DSB
ISB
```

761663

Wrong DFSR and IFSR descriptor format when taking an external abort from Non-secure to EL3 in AArch32

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

When an external abort that occurs on a translation table walk using the Long-Descriptor format is taken from Non-secure PL0 and PL1 to an EL3 Exception level using the Short-Descriptor format, the information related to the abort is reported into the Secure syndrome registers IFSR and DFSR using the Short-Descriptor format instead of the Long-Descriptor format. However, it is not possible to report an external abort that occurs on a Level 3 translation table walk using the Short-Descriptor translation table format. Moreover, the value of the fault status code that is held in the syndrome registers is UNKNOWN.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The erratum can be triggered in two cases, and in both cases the following conditions must be true:

- EL3 is in the AArch32 state.
- EL3 is using the Short-Descriptor translation table format (the Secure version of TTBCR.EAE is at 0).

Case 1:

- SCR.EA == 1
- HCR.VM == 1
- Non-secure PL0 and PL1 privilege modes are using the Short-Descriptor translation table format (the Non-secure version of TTBCR.EAE is at 0).
- An external abort occurs on stage 2 of an address translation in Non-secure PL0 or PL1 mode.

Case 2:

- EL1 is using the Long-Descriptor translation table format (the Non-secure version of TTBCR.EAE is at 1).
- An AT operation of the type ATS12NSO** is processed from EL3.
- An external abort occurs on stage 1 of the address translation.

Note: Using the Short-Descriptor translation table format when Non-secure PL0 and PL1 modes are using the Long-Descriptor format (as in Case 2) is not an expected use case.

Implications

- In Case 1, the Short-Descriptor translation table format is used for Secure DFSR and IFSR whereas the Long-Descriptor translation table format is recommended.
- In Case 2, the Short-Descriptor translation table format is used for Secure DFSR whereas the Long-Descriptor translation table format is expected.
- In both cases, when the abort occurs on a Level 3 translation table walk, the value reported in the Secure version of IFSR.FS for Case 1 and DFSR.FS for both cases is UNKNOWN. As a result, the value held by this field might match with a value that identifies another type of abort.

Workaround

There is no direct workaround for this erratum. However, under the conditions that might trigger this erratum:

- Software should only rely on the LPAE field of Secure registers DFSR and IFSR to identify the descriptor format used to report the exception.
- The value held in the FS field of Secure DFSR and IFSR should be considered carefully when taking to EL3 after a Data Abort or a Prefetch Abort. Indeed, the exception might be in fact an external abort that occurs on a Level 3 translation table walk.

771861

Access to ATCR_EL1 is not trapped when HCR.TVM or HCR.TRVM are set

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

In AArch64, accesses to ATCR_EL1 are not trapped when either HCR.TVM or HCR.TRVM is set.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered whenever ATCR_EL1 is accessed at EL1 Non-secure in AArch64 through an MSR or MRS instruction with either HCR.TVM or HCR.TRVM bit set.

Implications

The hypervisor is not informed of accesses to this register at EL1.

Workaround

Setting of HCR.TIDCP allows trapping of ATCR_EL1.

772905

Accesses to ETM registers trough the APB while the core is in Warm reset might hang

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

When the core is kept in Warm reset, for example in the case of an emulated powerdown, APB accesses to the ETM registers might hang.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions trigger this erratum:

- The core is kept in Warm reset.
- An APB access is performed to the ETM register space.

Implications

The APB access never gets a response, and hangs.

Workaround

Before accessing ETM registers through the APB, the debugger should force the Cortex-A75 core into debug state.

790748

Internal timing conditions might cause the CPU to stop processing interrupts

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

Depending on a set of internal timing conditions, the CPU might stop processing interrupts.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The erratum can only be hit with a particular setup in addition to a set of internal timing conditions:

- The CPU is executing in AArch32 state.
- The CPU attempts execution of a conditional MRC, MCR, WFE, or WFI instruction.
- This conditional instruction fails its condition code check.
- This conditional instruction is executed in the shadow of a mispredicted branch and is eventually flushed from the pipeline.

Implications

If the conditions described above are met, the CPU might stop processing interrupts until the execution of a Context Synchronization Operation

Workaround

Setting bit[13] of CPUACTLR_EL1 prevents this erratum.

Using Aarch32:

```
MRRC p15, 0, r0, r1, c15
ORR r0, r0, #1<<13
MCRR p15, 0, r0, r1, c15
ISB
```

Using Aarch64:

```
MRS X0, S3_0_C15_C1_0
ORR X0, X0, #1<<13
MSR S3_0_C15_C1_0, X0
ISB
```

787961

DRPS or Debug Exit with SCTLR_ELx.IESB set and a pending vSError causes erroneous behavior

Status

Affects: Cortex-A75
Fault Type: Programmer Category B
Fault Status: Present in r0p0.

Description

When SCTLR_ELx.IESB is set, an IESB is performed when executing a DRPS instruction or when leaving Debug state. If a vSError is pending and synchronized by the IESB when the IESB completes, the CPU fails to restore PSTATE from SPSR_ELx/DSPSR.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

- SCTLR_ELx.IESB is set to 1 for the Exception level currently in use by the CPU.
- A vSError is pending when the Debug exit or DRPS instruction is executed.
- The vSError would have been synchronized by the IESB if the CPU had not been in Debug state.

Implications

When the above conditions are met when exiting Debug State, the CPU leaves Debug state on completion of the IESB without restoring PSTATE from DSPSR:

- PSTATE.{IT, T} are ignored (because this is caused by IESB, the CPU is always executing in Aarch64 state following the Debug Exit)
- PSTATE.SS is UNPREDICTABLE
- PSTATE.{D, A, I, F} are all 1
- all other PSTATE fields retain the value used during execution in Debug State

When the above conditions are met when executing a DRPS instruction, the affected DRPS instruction behaves as a NOP.

Workaround

There is no workaround.

Category B (rare)

739639

Erroneous allocation of Page Table Entries might lead to a conflict abort

Status

Affects: Cortex-A75
Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0.

Description

Under certain conditions, which include the ETM stalling the Cortex-A75 processor, the Instruction micro-TLB might allocate a translation already cached within that structure, leading to the report of a conflict abort on a subsequent access to the same page. The conflict abort is not caused by an erroneous software management of page table entries.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

1. The Cortex-A75 processor switches from a translation regime to a new one.
2. The instruction fetch sends a translation request to the MMU during the switch to the new translation regime.
3. When the translation comes back from the MMU, the ETM stalls the processor.

Implications

The returned translation might already be present in the Instruction micro-TLB, possibly leading to a translation entry being allocated twice. On a subsequent lookup hitting in the corresponding page, a conflict abort will get reported.

Workaround

If the ETM is not enabled, no workaround is required as the erratum cannot be triggered.
If the ETM is enabled, a TLB invalidation of the faulty address, as reported in IFAR, in the context the abort was taken in, solves the issue.

Category C

869960

A PMU snapshot request returns erroneous value when PMU counters are disabled

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: present in r0p0, r1p0, r1p1

Description

On a PMU snapshot request when the PMU counters are disabled, the read value is corrupted.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions are required to trigger this erratum:

- The PMU counters are disabled.
- A snapshot request is initiated.

Implications

A read of the PMU snapshot registers will return an erroneous value.

Workaround

This erratum can be worked around by setting the PMU counters enable bit (PMCR_EL0.EN) to 1 before triggering a PMU snapshot request.

858572

Vector catch can be generated to EL1 AArch64

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1. Open.

Description

Executing an instruction in EL0 AArch32 using the AArch64 stage 1 translation regime at an exception vector address (VBAR+0x0..0x1C) which is programmed for address matching in DBGVCR will trigger an unexpected Vector Catch.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

- The processor is executing in EL0 AArch32 using the AArch64 EL1 translation regime.
- The processor is executing an instruction at the address of an exception (VBAR+0x0..0x1C) vector which is programmed for address matching in DBGVCR.

Implications

The processor generates a Vector Catch exception to EL1 AArch64 which is unexpected in the programmers' model.

Workaround

To avoid this erratum, software can write zero to DBGVCR32_EL2 before switching to EL1 using AArch64.
Note 1: Use of Vector Catch is deprecated.
Note 2: In a platform operating system, executing an instruction in EL0 AArch32 using the AArch64 stage 1 translation regime at an exception vector address (VBAR+0x0..0x1C) should result in an MMU fault, leading to an Instruction Abort exception which has priority over the Vector Catch exception.

815950

Using a 52-bit translation table base address does not report an address size fault

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1.

Description

When software is configuring a translation stage to use a translation table base address with a 52-bit Physical Address (PA), the processor does not report an address size fault. Consequently, the processor might use an incorrect translation table base address.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered if the following conditions are true:

- The processor is in AArch64 state.
- For one of the stage 1 translations:
    - The 64KB translation granule is used for this stage (TCR_EL1/2.TG0/TG1 == 0b11 or TCR_EL2/3.TG = 0b11).
    - The corresponding Translation Table Base Register of this stage is holding a 52-bit PA (the corresponding register TTBR*_EL1/2/3 holds non-zero address bits in bits[5:2]).
    - The first level of lookup on this translation stage is processed by the MMU.
- For the stage 2 translation:
    - The 64KB translation granule is used (VTCR_EL2.TG == 0b11).
    - The corresponding Translation Table Base Register of this stage is holding a 52-bit PA (the corresponding register VTTBR_EL2 holds non-zero address bits on bits[5:2]).
    - The first level of lookup on this translation stage is processed by the MMU.

This applies to every supported PA and Intermediate Physical Address (IPA) size (programmed in TCR_EL2/3.PS or VTCR_EL2.PS for the PA size and in TCR_EL1/2.IPS for the IPA size).

Implications

- No address size fault is generated. As a result, software is not informed that it attempted to use an incorrect PA or IPA for the corresponding translation table base address.
- Values held in bits[5:2] of the incorrect translation table base address are treated as if they had a value of 0. Consequently, software mught use an incorrect translation table base address. While a translation table base address such as ADDR[51:0] is expected to be used, the translation table base address that is programmed will be {0b0000, ADDR[47:0]}.
- The value read back in bits[5:2] of registers TTBR*_EL1/2/3 and VTTBR_EL2 might be either the value written or 0.

Workaround

Software must ensure bits[5:2] of any translation base address register being written are 0.

801389

Accesses to some cluster Performance Monitor system registers are Read-As-Zero, Writes Ignored

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r1p0, r1p1.

Description

Accesses from the Cortex-A75 processor to cluster Performance Monitor system registers CLUSTERPMMDCR in AArch32, CLUSTERPMMDCR_EL3 in AArch64, or CLUSTERPMCCNTR using the 64-bit access in AArch32 are corrupted. Writes to these registers using these accesses are ignored, and reads return the value 0. Therefore, the counting of Secure cluster events cannot be enabled and bits[63:32] of CLUSTERPMCCNTR are not accessible in AArch32.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered when one of the following conditions is true:

- The Cortex-A75 processor is in AArch32 state and one of the following system registers is accessed:
    - CLUSTERPMCCNTR using MRRC/MCRR instructions.
    - CLUSTERPMMDCR using MRC/MCR instructions.
- The Cortex-A75 processor is in AArch64 state and CLUSTERPMMDCR_EL3 is accessed using MRS/MSR instructions.

Implications

The registers are treated as Read-As-Zero, Writes Ignored. Therefore, the counting of Secure cluster events cannot be enabled for cluster performance monitoring and for registers used by software to inform decisions about L3 partial cache power-down like CLUSTERL3HIT and CLUSTERL3MISS.

Workaround

- The cluster system registers are shared among the cores within the same FCM. Therefore, accesses to registers CLUSTERPMMDCR, CLUSTERPMMDCR_EL3, and 64-bit access to CLUSTERPMCCNTR might be performed from another core within the same FCM.
- If the FCM contains only Cortex-A75 cores:
    - There is no workaround for accesses to CLUSTERPMMDCR and CLUSTERPMMDCR_EL3.
    - There are three possible alternatives to access CLUSTERPMCCNTR:

1. The lower 32 bits remain accessible using MRC/MCR instructions:

```
MRC p15, 0, <Rt>, c15, c6, 0
MCR p15, 0, <Rt>, c15, c6, 0
```

2. The full register remains accessible from higher Exception levels in AArch64 using MRS/MSR instructions to access CLUSTERPMCCNTR_EL1:

```
MRS <Xd>, S3_0_c15_c6_0
MSR S3_0_c15_c6_0, <Xd>
```

3. Writing 1 in CLUSTERPMCR(_EL1).C resets the full CLUSTERPMCCNTR to zero.

770356

Infinite loop of Store instructions might prevent older STR instruction from completing

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1. Open.

Description

An infinite loop of Store instructions might prevent older, in program order, STR instructions from being executed, leading to a potential QoS problem.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

- Execution of older STR instructions must be delayed, waiting for operands to be available.
- While these instructions are stalled, the infinite loop of Store instructions starts executing and never gets interrupted.

Implications

The older STR instruction might never become visible to external observers until the continuous loop is interrupted.

Workaround

Infinite loops of Store instructions should be avoided. If such a loop is meaningful, a DMB should be executed before the infinite loop.

1058157

Incorrect ETM timestamp value when timestamp and event generation happen on same cycle

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault: Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1. Open.

Description

When an event packet and timestamp packet are generated on the same cycle, the value of the timestamp packet should be sampled at the time of that event. Because of this erratum, the value of the timestamp is sampled on the previous atom or event packet.

Configurations Affected

All configurations are affected.

Conditions

- Timestamping is enabled.
- Event packet generation is enabled.
- An event packet and a timestamp packet must be generated on the same cycle.

Implications

The timestamp value might be slightly out of date. In a typical system, atom packets and event packets are frequent relative to the global timestamp signal increment, which means that the amount of errors will be small.

Workaround

There is no workaround for this erratum.

1058155

Cycle count value in timestamp packet might be incorrect

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault: Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1. Open.

Description

When a timestamp packet is generated in the trace, the timestamp packet can indicate the cycle count between the previous cycle count element and the element the timestamp is associated with. This can be used to infer the alignment between cycle count elements and the global timestamp. Timestamp packets can be inserted in the trace stream some time after the element the timestamp is associated with gets traced. Because of this erratum, the cycle count indicated in the timestamp packet will be determined by the time when the packet is inserted in the trace stream.

Configurations Affected

All configurations are affected.

Conditions

- Timestamping must be enabled, with TRCCONFIGR.TS== 1.
- Cycle counting must be enabled, with TRCCONFIGR.CCI== 1.

Implications

The cycle count for the timestamp packet will indicate a time after the element which was used to capture the timestamp. If there has been a gap in the tracing of elements which can be timestamped, this error can be large. This does not affect the incremental cycle count values which are related to instructions being committed.

Workaround

There is no workaround for this erratum.

836130

Corrected Error Counter Repeat is incremented when detecting any first error

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1, open

Description

When detecting a first error, the Corrected Error Counter Repeat of the error record, ERR0MISC0.CECR, is always incremented even if the detected error is not a Corrected Error (CE). Therefore, the number of CEs reported in ERR0MISC0.CECR is one higher than expected if the first error is a Deferred Error (DE) or an Uncontainable error (UC). If the ERR0MISC0.CECR was initialized at its maximal value, because of this erratum it overflows and a Fault Handling Interrupt might be triggered if the interrupt is enabled for CEs.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered when:

- No error is recorded on the Error Record Registers (ERR0STATUS.V is cleared).
- A DE or UC is detected.

Implications

The value held in ERR0MISC0.CECR is incremented.
If this value was programmed to 0b1111111, ERR0MISC0.CECR overflows and consequently:

- Bits ERR0STATUS.OF and ERR0MISC0.OFR are set.
- If Fault Handling Interrupt is enabled for CEs (ERR0CTLR.CFI is set), a Fault Handling Interrupt is triggered even if Fault Handling Interrupt is disabled for DEs and UCs.

Workaround

If no CE has been detected (ERR0STATUS.CE == 0b00), then software can assume that the value held in ERR0MISC0.CECR is one higher than expected. Otherwise, it is not possible to know if this value is the right one or one higher than expected.

820018

Exceptions taken from AArch32 to AArch64 might generate an Address Size Fault when the corresponding VBAR_ELx is using a tagged address

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1.

Description

An exception taken from AArch32 to AArch64 might generate a fault when VBAR_ELx, used as the exception target, is using tagged addresses (the corresponding TCR_ELx.TBI bit is set and bits [63:56] of VBAR_ELx are different from bit[55]).
If Stage1 translation is enabled, then the exception might generate a Translation Fault. If Stage1 translation is disabled, then the exception might generate an Address Size Fault.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions are required to trigger this erratum:

- TCR_ELx.TBI is set for the EL the exception is taken to.
- VBAR_ELx[63:56] differ from VBAR_ELx[55] for the EL the exception is taken to.

Implications

Instead of executing the exception normally, because VBAR[63:56] is not a sign-extension of VBAR[55], the exception might generate either:

- A Translation Fault, if Stage1 translation is enabled.
- An Address Size Fault, if Stage1 translation is disabled.

Workaround

There is no workaround.

900547

ESB in Debug state does not defer SErrors

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1; open.

Description

In Debug state, ESB instructions execute as NOPs.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum might be triggered when executing an ESB in Debug state.

Implications

This erratum might cause a pending SError to remain pending instead of being deferred by the execution of an ESB instruction.

Workaround

There is no workaround.

877067

The dirty state bit of a translation table descriptor might erroneously be updated

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1.

Description

Under certain timing conditions, the dirty state bit of the associated translation table descriptor might get speculatively updated.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum occurs when the following conditions are met:
1. A cacheable atomic instruction is being executed by the processor.
2. A following store in program order which, if architecturally performed, would update the dirty state bit of the translation table descriptor, is executed. The associated Store access is not performed, pending completion of the previous atomic instruction.
3. The atomic instruction replays because it cannot ensure atomicity of the memory access.

Note that these 3 conditions have to happen with particular timing conditions.

Implications

The dirty state bit of the translation table descriptor is set whereas the store has not been executed, which implies that software could see the page marked as dirty whereas it is not.

Workaround

Software has to be resilient to a page which translation table descriptor is marked as dirty while it only contains clean data.

872089

SError taken as a result of stepping an ESB instruction might incorrectly set EDESR.SS

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0 and r3p1.

Description

An SError that is taken as a result of being synchronized by an ESB instruction that is executed in halting-step might cause EDESR.SS to be set, whereas it should not be set in that case.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions are required to trigger this erratum:

- The synchronized SError is taken to EL3.
- ExternalSecureInvasiveDebugEnabled() is FALSE.

Implications

EDESR.SS might be spuriously set. This means that after performing an ERET to resume hardware stepping, the processor might immediately reenter debug step without stepping any instruction.

Workaround

There is no workaround.

832595

EDSCR.INTdis might erroneously cause asynchronous exceptions routed to EL2 to be masked

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: present in r0p0, r1p0, r1p1, r2p0, r2p1, r3p0, and r3p1. Open.

Description

When EDSCR.INTdis == 0x1 while HCR_EL2.TGE and HCR_EL2.E2H are both set, an asynchronous exception at EL0 that would normally be taken to EL2 might not be taken and remain pending.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The erratum can happen under the following conditions:

- EDSCR.INTdis == 0x1.
- ExternalInvasiveDebugEnabled() is TRUE.
- HCR_EL2.E2H is set.
- HCR_EL2.TGE is set.
- The CPU is executing at EL0.

Implications

Under the above conditions, an asynchronous exception received while executing at EL0 will remain pending instead of being taken to EL2.

Workaround

There is no workaround.

792334

PSTATE.UAO might affect LDR*T/STR*T instructions executed in AArch32

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When an exception return to AArch32 is performed with SPSR_ELx.UAO set to 1, this value is propagated to PSTATE.UAO and subsequently affects the behavior of LDR*T and STR*T instructions executed in AArch32.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

An exception return to AArch32 is performed with SPSR_ELx.UAO set to 1.

Implications

In the configuration described above:

- The behavior of AArch32 LDR*T/STR*T instructions executed in EL0 is unchanged.
- LDR*T/STR*T instructions executed in an EL using AArch32 other than EL0 will behave as the equivalent LDR*/STR* instructions.
  This erratum has no impact on AArch64.

Workaround

The workaround is to sanitize the SPSR value when returning to AArch32 in order to ensure SPSR_ELx.UAO is not set when the mode being returned to is AArch32.

792226

Leaving Debug state from EL0 does not restore PSTATE.PAN and PSTATE.UAO from DSPSR

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When the CPU exits Debug state while executing at EL0, PSTATE.PAN and PSTATE.UAO are not restored from DSPSR. Instead, they retain the value that was used while in Debug state.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

To trigger this erratum the Debug exit must be performed from EL0.

Implications

PSTATE.UAO and PSTATE.PAN will retain the value used during execution in Debug State.
Note:

- It is illegal to use a Debug Exit from EL0 to return to an EL other than EL0
- Since PSTATE.PAN and PSTATE.UAO have no effect on EL0 execution, this does not affect EL0 behaviour
- PSTATE.PAN and PSTATE.UAO will be set to their expected value by an exception.
- The only visible effect is that upon re-entry in Debug State or execution of an exception, SPSR_ELx.UAO and SPSR_ELx.PAN will have unexpected values.

Workaround

To avoid this erratum, the debugger can perform a DRPS operation to EL0 that sets PSTATE.UAO and PSTATE.PAN to the expected value. This will ensure that these flags have the expected value after exiting Debug State from EL0

782841

Alignment fault on ERET might report an incorrect address tag in IFAR

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When a PC alignment fault is generated as a result of executing an ERET instruction with a misaligned ELR_ELx, the IFAR might report an incorrect address tag.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The erratum might be triggered if:

- The processor is running in AArch64 state.
- An ERET instruction is executed, switching from ELx to Ely.
- The applicable TCR_ELy.TBI bit has a different configuration than the TCR_ELx.TBI bit.
- A PC alignment fault is triggered as a result of executing the ERET instruction.

Implications

In this case, the top eight bits of IFAR will contain one of the following values:

- If TCR_ELx.TBI0/1 for the source EL is set, the top eight bits of IFAR will be incorrectly sign-extended based on bit[48] of the Virtual Address instead of reporting the tag address.
- if TCR_ELx.TBI0/1 for the source EL is cleared, the top eight bits of IFAR will incorrectly report the tag address instead of being sign-extended.

Workaround

There is no workaround.
Note that ARM does not expect an alignment fault to be triggered upon executing an ERET instruction.

782705

Aborts on AT operations are ignored in Debug state

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When the processor is in Debug state, aborts affecting an AT operation might not be reported. As a result, neither the syndrome register nor the PAR(_EL1) are updated, and the exception is not taken.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

- The processor is in Debug state.
- An AT operation is executed.
- An abort is generated affecting the AT operation, either because of an External Abort or a stage 2 abort on a stage 1 page table walk for a VA to IPA translation operation.

Implications

- The processor behaves as if the AT operation was completed.
- No abort abort is taken.
- The processor does not switch to the appropriate Exception level to handle the abort.
- The appropriate syndrome registers (ESR_ELx, DFSR, HSR) are not updated.
- The PAR(_EL1) is not updated.

Workaround

A debugger can use the following sequence to detect that the AT operation aborted and work around this erratum:

- Write a value on PAR(_EL1) register which cannot be returned by an AT operation, such as 0x00000001.
- Execute the AT operation
- Read the value held by PAR(_EL1) register.

If the PAR(_EL1) value is still equal to 0x00000001, this means that the AT operation should have aborted.

783218

DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might be erroneous

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When the ExternalSecureNonInvasiveDebugEnabled condition is true, DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might get corrupted.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

- The ExternalSecureNonInvasiveDebugEnabled condition is true.
- The value of MDCR_EL3.SPME, if the processor is executing in AArch64, or the value of SDCR.SPME, if the processor is executing in AArch32, is 0.

Implications

DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis are set to 0, whereas they should be set to 1.

Workaround

Use of these fields is deprecated. Therefore, ARM recommends that you do not use these fields.

764571

PC Sample register and Context ID registers are not consistent

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

Performance Monitors Program Counter Sample Register PMPCSR is sampled at a different time than context sample registers PMCID1SR, PMCID2SR, and PMVIDSR.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered whenever the processor reads the PC Sample Registers defined in the optional PC Sample-based Profiling extension.

Implications

The value of PMPCSR is inconsistent with the context read from PMCID1SR, PMCID2SR, and PMVIDSR. All registers still hold true sampled values, but are sampled at a different time.

Workaround

This erratum can be avoided by setting bit 4 of register CPUACTLR2_EL1.

```
MRS X0, S3_0_C15_C1_1
ORR X0, X0, #1<<4
MSR S3_0_C15_C1_1, X0
ISB
```

ARM does not advise applying this workaround by default on production devices; a small power increase might happen, in the range of 0.5%.

767916

An exclusive sequence encountering poisoned data might not be able to progress

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

Under certain timing conditions, if an LDXP instruction using 64-bit registers continuously encounters poisoned data, then the associated memory access might infinitely replay.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor implementing ECC.

Conditions

The following conditions might trigger this erratum:
1. An exclusive sequence is performed on a 128-bit region accessed by an LDXP instruction using two 64-bit registers.
2. Exactly one half of this 128-bit region contains poisoned data.
3. The LDXP instruction hits in the L1 cache.

Implications

The LDXP instruction might infinitely replay in the core without reporting a System Error Interrupt.

Workaround

There is no workaround for this erratum but the core would still be able to process interruptions, provided they are not masked.

780153

MSB of PMPCSR does not report sign extension of the Program Counter

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

Bits[55:49] of the PC Sample Register PMPCSR defined in the optional PC Sample-based Profiling extension always read 0, instead of reporting the sign extension of the sampled PC.

Configurations affected

This erratum affects all configuration of the Cortex-A75 processor

Conditions

This erratum is triggered whenever the PMPSCR is read.

Implications

The value of PMPCSR[55:49] always reads zero, even when bit[48] of the PC is 1.

Workaround

When reading the PMPCSR, software might extend bit[48] of the PMPCSR to bits[55:49].

771824

DBGDSCRext.MOE and DBGDSCRint.MOE might get corrupted when entering Debug state

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When entering Debug state on a debug event in AArch32 state, the DBGDSCRext.MOE and DBGDSCRint.MOE registers are updated.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The processor is entering Debug state.

Implications

DBGDSCRext.MOE and DBGDSCRint.MOE might get corrupted if used by an external debugger.

Note: Self-hosted debug and external debug are unlikely to be used at the same time.

Workaround

There is no workaround.

772669

Single stepping might fail to step an instruction

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

The processor can enter Debug state without executing a single instruction in any of the following cases:

- The processor returns from Debug state to an Exception level where Halting is allowed, and the Halting step state machine is in Active-not-pending state.
  Or
- Halting is allowed, the Halting step state machine is in Active-not-pending state, and the stepped instruction is trapped to an Exception level where Halting is prohibited.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered in any of the following cases:

- A Load Multiple (LDM) instruction that updates the PC is single stepped.
- The instruction that is single stepped is trapped to an Exception level where Halting is prohibited.

Implications

Because of this erratum, on the subsequent single step, the processor could re-enter Debug state without executing any single instruction.

Workaround

Single stepping a further instruction will work.

774033

External changes to EDSCR.HDE might lead to unauthorized exception generation

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

Changing EDSCR.HDE using an external agent might cause the CPU to fail to enter Debug state when hitting the enabled breakpoint, vector catch, or watchpoint.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

- EDSCR.SDD is 0, or the CPU is in Non-secure state.
- EDSCR.HDE is set by an external agent to enable either a breakpoint, a vector catch, or a watchpoint.
- The programmed breakpoint, vector catch, or watchpoint is hit before a Context Synchronization Operation is performed.

Implications

In these conditions, the CPU might route the generated exception as if EDSCR.HDE was not set instead of entering Debug state:

- If the CPU is executing in Non-secure state, the exception will be taken to EL1 or EL2 depending on the Instruction Abort/Data Abort exception routing rules.
- If the CPU is executing in Secure state with EDSCR.SDD=0, the exception will be taken to EL3 if the CPU is in EL3, otherwise to EL1.

Workaround

The EDSCR.HDE should only be set while the processor is in Debug state.

769222

Error not correctly recorded if reported when software is clearing ERR0STATUS

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When fields ERR0STATUS.{V, AV, ER, MV, SERR} of the CPU Error Record Primary Status Register ERR0STATUS are being cleared and an error with a priority lower than or equal to the error currently recorded is reported simultaneously, these fields are corrupted because they remain cleared but don't record the information about the new error. Therefore, the record of the new error might not raise the expected Fault Handling Interrupt (FHI) or Error Recovery Interrupt (ERI). Moreover, subsequent errors may not overwrite error records even if, regarding the corrupted value held by register ERR0STATUS, the values held by error record registers are reported as not valid (field ERR0STATUS.V is cleared). However, the other fields of ERR0STATUS (including CE, DE, or UE) remain correctly updated when recording subsequent errors.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered if the following conditions are true:

- An error (called error A) was previously recorded.
- A direct write of the processor clears one of the fields ERR0STATUS.{V, AV, ER, MV, OF, SERR}.
- At the same cycle, a new error (called error B), with a priority lower than or equal to the priority of error A, is detected and recorded.

Implications

- Fields ERR0STATUS.{V, AV, ER, MV, OF, SERR} are corrupted. These fields are cleared by the write but don't reflect the record of error B.
- Registers ERR0MISC0 and ERR0ADDR are corrupted. These registers remain unchanged and don't reflect the record of error B.
- Fields ERR0STATUS.{UE, CE, DE} remain correct. They hold the values that are expected after the write access, followed by the record of error B.
- Because field ERR0STATUS.V is corrupted and may be incorrectly cleared:
    - Software might assume that no error has been recorded since error A.
    - If FHI is enabled (at least one of the ERR0CTLR.FI, CFI fields is set), it is never triggered when it receives error B.
    - If ERI is enabled (field ERR0CTLR.UI is set), it is never triggered when it receives error B.
- When detecting any error (called error C) following error B:
    - If error C priority is greater than error B, the error record is overwritten
    - Otherwise, fields ERR0STATUS.{V, AV, ER, MV, SERR} remain corrupted and unchanged, but fields ERR0STATUS.{UE, CE, DE} are correctly updated

Workaround

Right after writing on register ERR0STATUS, re-read the value held by the register. Information held in fields ERR0STATUS.{UE, CE, DE} remains valid even in the faulting case (and even if bit ERR0STATUS.V is cleared). Particularly, if an Uncorrected Error (that is, an Uncontainable error in the Cortex-A75 processor) occurs at any time between the two read accesses:

- For an Uncontainable error that occurs before the software write, the behavior remains in accordance with the architecture (and ERR0STATUS.UE is set).

- For an Uncontainable error that occurs during or after the software write, ERR0STATUS.UE is set even if the write was supposed to clear it.
  This can be used as a workaround to the Error Recovery Interrupt which is not triggered in the faulting case.

790861

IESB might fail to synchronize a pending SError

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

IESB might fail to synchronize a pending SError.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The erratum can be hit in the following software configurations:

If executing in EL3:

- SCR.EA is initially clear.
- SCR.EA is then set without performing a Context Synchronization operation.
- No other Context Synchronization is performed between the change to SCR.EA and ERET.

if executing in EL2:

- HCR.AMO is initially clear.
- HCR.AMO is then set without performing a Context Synchronization operation.
- No other Context Synchronization is performed between the change to HCR.AMO and ERET.
- SCR.EA is clear (SError is configured to be taken to EL2).

Implications

If the conditions described above are met, the implicit ESB caused by the first ERET instruction following the change to SCR.EA or HCR.AMO might fail to properly synchronize a pending SError.

Workaround

There is no workaround for this erratum

754852

EDPRCR.CORENPDRQ might not get set after a Cold reset

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

On a Cold reset, EDPRCR.CORENPDRQ is always reset to 0 instead of reflecting EDPRCR.COREPURQ.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The processor is leaving the powerdown state with EDPRCR.COREPURQ set.

Implications

The values configured for EDPRCR.CORENPDRQ when the processor is in the powerdown state will not be correctly reflected after the powerup. As a result, if the debugger subsequently clears EDPRCR.COREPURQ, the next powerdown will not be emulated.

Workaround

The debugger must keep EDPRCR.COREPURQ set. If the debugger needs to clear EDPRCR.COREPURQ, for example because it is delegating control over emulated powerdown to software, it must halt the processor after powerup and set the EDPRCR.CORENPDRQ bit when the processor is in Debug state.

754402

ESR, IFSR, and HSR might get corrupted when entering debug state

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

Debug state entries can corrupt ESR_ELx, IFSR, and HSR of the EL the processor has halted in.

When entering debug state on a breakpoint debug event:

- In AArch64, the ESR_ELx register of the current EL is updated.
- In AArch32, the IFSR register, if the current EL is EL1, or the HSR register, if the current EL is EL2, is updated.

The corrupted value corresponds to the value the register would have been updated to if breakpoint had been taken as a Breakpoint debug exception, for example, if using self-hosted debug.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The processor is entering debug state on breakpoint debug events.

Implications

The registers mentioned above get updated as if the debug exception had been triggered in self-hosted debug.

Workaround

An external debugger must not set hardware breakpoints in the critical section of exception handlers before these registers are saved, nor in the tail of the handlers after these have been restored before the exception return. To halt execution at the start of an exception handler, before these registers are saved, the Exception Catch debug event should be used instead of hardware breakpoints.

753282

Executing an ESB instruction in Software Step might cause a synchronized SError to be deferred instead of taken

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

Depending on timing conditions, if an SError is synchronized as a result of the execution of an ESB while in Software Step, then this SError might be deferred instead of taken.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions might trigger this erratum when an ESB instruction executes while in Software Step:

- SCTLR_ELx.IESB is 0 for the EL executing the ERET instruction that causes entry into active-not-pending state.
- The SError that is synchronized by the ESB is not masked at EL_D.

Implications

The synchronized SError might be deferred instead of taken.

Workaround

Setting SCTLR_ELx.IESB prevents this erratum from occurring.

751005

An MCR to ERXFR2 register in non-User mode does not generate an UNDEFINED exception

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

At EL1 or above, a write to the ERXFR2 register does not generate an UNDEFINED exception.
Note that the write is ignored.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum is triggered when executing an MCR to ERXFR2 register at EL1 or above.

Implications

Because of this erratum, any write access to this register is not UNDEFINED in AArch32. Instead, the MCR is :

- Write-ignored when HCR.TERR = 0 and SCR.TERR = 0.
- Trapped to monitor when HCR.TERR = 0 and SCR.TERR = 1.
- Trapped to hypervisor when HCR.TERR = 1.

Workaround

No workaround is expected for this erratum.

750234

ETM might trace an incorrect branch PC on taking an SError synchronized by an ESB

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When an SError is synchronized and taken as a result of executing an ESB instruction, the ETM reports the branch PC as if the ESB instruction had not synchronized an SError.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

This erratum might be triggered after the execution of an ESB instruction that synchronizes an SError.

Implications

Although the ESB instruction successfully synchronized the SError, the PC value does not show this. Instead, the PC value shows the correct value plus 4, as if the SError was not synchronized.

Workaround

There is no workaround for this erratum.

739130

Inversion in Trap priority between ICH_HCR.TALL0/TALL1 and SCR.FIQ/IRQ

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

If ICH_HCR.TALL0 is 1, HCR.FMO is 0, and SCR.FIQ is 1, then access to GIC registers of group 0 in Non Secure EL1 causes either of an Undef or a trap to EL3 instead of trap to EL2.
If ICH_HCR.TALL1 is 1, HCR.IMO is 0, and SCR.IRQ is 1, then access to GIC registers of group 1 in Non Secure EL1 causes either of an Undef or a trap to EL3 instead of trap to EL2.

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions trigger this erratum:

- ICH_HCR.TALL0 is 1, HCR.FMO is 0, and SCR.FIQ is 1.
- One of the following registers is accessed in Non Secure EL1: ICC_IAR0, ICC_EOIR0, ICC_HPPIR0, ICC_BPR0, ICC_AP0R0, or ICC_IGRPEN0.

Or

- ICH_HCR.TALL1 is 1, HCR.IMO is 0, and SCR.IRQ is 1.
- One of the following registers is accessed in Non Secure EL1: ICC_AP1R0, ICC_IAR1, ICC_EOIR1, ICC_HPPIR1, ICC_BPR1, or ICC_IGRPEN1.

Implications

Because of this erratum, any access to one of the registers listed above is trapped to EL2 instead of being either:

- Undefined, if EL3 is AArch32.

Or

- Trapped to EL3, if EL3 is AArch64.

Workaround

If EL3 is AArch64,then the EL3 can emulate the trap correctly.
There is no workaround if EL3 is AArch32.

770802

ETM might not generate an event packet and ATB trigger

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

The ETM contains four resources to generate events based on core activity. When the ETM generates an event, it is reported to the CTI on the external outputs bus, an ATB trigger is generated, and an event packet is inserted into the trace stream.

Because of this errata, when the ETM is becoming idle, there is a one cycle window where an event might get indicated to the CTI, but would not generate an ATB trigger and would not generate an event packet.

Configurations Affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

1. The ETM is becoming idle due to any of the following
    - The ETM has been disabled due to TRCPRGCTLR.EN = 0 or TRCOSLAR.OSLK = 1
    - The core has started to execute a WFI or WFE and is going to enter sleep
    - **DBGEN**/**NIDEN** have changed and trace is no longer permitted
2. The resources are configured to generate events.
3. An event is generated because of a PMU event or CTI trigger on the last cycle in which the ETM could generate an event.

Implications

If the stimulus that caused the event to be generated occurred one cycle later, then the event would not have been generated at all. Therefore, this errata will only be noticed when trying to correlate the CTI behaviour with the trace stream.

Workaround

There is no workaround for this erratum.

770785

ETM might assert AFREADY before all trace has been output

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

When the **AFVALID** signal on the ATB interface is asserted, the ETM should immediately start outputting all buffered trace. It should assert the **AFREADY** output one cycle after all trace that was buffered on the cycle in which **AFVALID** was first asserted has been output.

Because of this erratum, the **AFREADY** signal might be asserted before all the necessary trace has been output.

Configurations Affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

1. The ETM must contain buffered trace.
2. **ATVALID** must be low on the first cycle **AFVALID** is high.

Implications

This might result in the ETM containing trace that was generated before the flush request when the rest of the system expects this trace to have been output.

Workaround

The system can ensure that all trace has been drained from the ETM by disabling it. The ETM can be disabled by setting TRCPRGCTLR.EN to 0. The system should then poll the TRCSTATR.IDLE bit. When it reads as 1, the ETM is idle and all trace that was generated before the write to TRCPRGCTLR has been output.

764081

Implicit Error Synchronization Barrier might not be correctly generated

Status

Affects: Cortex-A75
Fault Type: Programmer Category C
Fault Status: Present in r0p0.

Description

On an exception taken from ELx to ELy, with x differing from y (x < y), an implicit Error Synchronization Barrier (ESB) is inserted after the exception if and only if the SCTLR_ELx.IESB associated to the EL the exception is taken from is set, rather than if and only if the SCTLR_ELy.IESB associated to the EL the exception is taken to is set, .

Configurations affected

This erratum affects all configurations of the Cortex-A75 processor.

Conditions

The following conditions might trigger this erratum:

- SCTLR_ELx.IESB is 0 for the EL the exception is taken from.
- SCTLR_ELy.IESB is 1 for the EL the exception is taken to.

Implications

If SCTLR_ELy.IESB is 1 and SCTLR_ELx.IESB is 0, it is not guaranteed that all synchronizable errors generated before the exception taken to ELy have pended an SError interrupt exception. As a result, the error might be taken later causing the error to be misattributed to ELy. However, the error is not lost. If the error recovery software would otherwise have attempted to recover from the SError interrupt (for example, by terminating software executing at ELx and recovering software executing at ELy), then this erratum might lead to a small increase in the detected uncorrected error failure rate (DUE FIT) for ELy.

For systems where SError interrupts are routed by software to EL3, the implicit ESB is not recommended for exceptions taken to EL3 because SError interrupts are always masked on entry to EL3. ARM expects such software to use the explicit ESB instruction instead, which is unaffected by this erratum.

Workaround

Implicit ESB should be enabled at all Exception levels.