Cortex-A55 Processor MP070

Software Developer Errata Notice

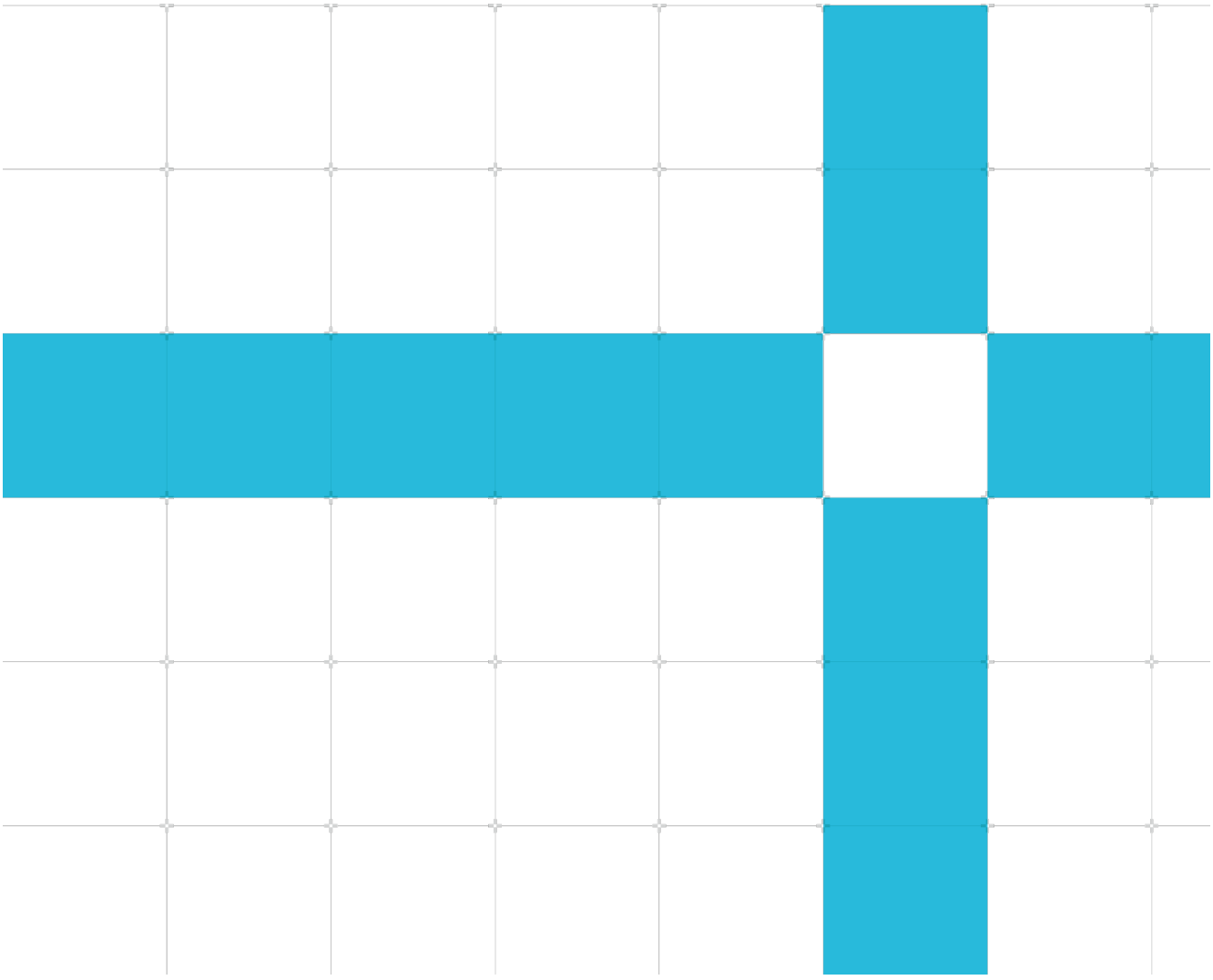
Date of issue: 15-Jun-2022

Non-Confidential

Document version: v15.0

Document ID: SDEN-859338

This document contains all known errata since the r0p0 release of the product.



Non-confidential proprietary notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(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.

Unrestricted Access is an Arm internal classification.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Cortex-A55 Processor MP070, create a ticket on **https://support.developer.arm.com**.

To provide feedback on the document, fill the following survey: **https://developer.arm.com/documentation-feedback-survey**.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

If you find offensive language in this document, please email **terms@arm.com**.

Contents

r0p1 implementation fixes

Note 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]	827496	Data cache maintenance operations by set/way targeting a 4MB L3 cache might affect incorrect cache index

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

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** identifies errata that have been fixed in each product revision.

15-Jun-2022: Changes in document version v15.0

ID	Status	Area	Category	Summary
2441007	New	Programmer	Category B (rare)	Completion of affected memory accesses might not be guaranteed by completion of a TLBI
2109978	New	Programmer	Category C	Atomic store instructions might not report an External abort or System Error
2288055	New	Programmer	Category C	DTR flags not cleared on external debugger access while leaving Debug state
2342475	New	Programmer	Category C	Halting step syndrome might be wrong on stepping a Load-Exclusive instruction
2364958	New	Programmer	Category C	Some unallocated debug and trace System registers might be trapped by HCR_EL2.TIDCP

02-Oct-2019: Changes in document version v14.0

ID	Status	Area	Category	Summary
1530923	New	Programmer	Category B	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation
1131793	Updated	Programmer	Category C	Stores might prevent forward progress of an exclusive loop
1401736	New	Programmer	Category C	Non-cacheable loads of mismatched size might not be single-copy atomic

20-Nov-2018: Changes in document version v13.0

ID	Status	Area	Category	Summary
1221012	New	Programmer	Category B (rare)	A load exclusive following a store or atomic to the same memory location can result in a lack of forward progress
1131793	New	Programmer	Category C	Stores might prevent forward progress of an exclusive loop
1214720	New	Programmer	Category C	VFP_SPEC PMU event might count incorrectly

09-Feb-2018: Changes in document version v12.0

ID	Status	Area	Category	Summary
1030596	New	Programmer	Category C	Cycle count value in timestamp packet might be incorrect
1030597	New	Programmer	Category C	Incorrect ETM timestamp value when timestamp and event generation happen on same cycle

15-Dec-2017: Changes in document version v11.0

ID	Status	Area	Category	Summary
1024718	New	Programmer	Category B	Update of DBM or AP bits without break-before-make might result in incorrect hardware dirty bit update
1017312	New	Programmer	Category C	DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might be erroneous

15-Nov-2017: Changes in document version v10.0

ID	Status	Area	Category	Summary
948532	New	Programmer	Category C	Incorrect PMCEID1 value
999993	New	Programmer	Category C	Multiple concurrent ECC errors might cause silent data corruption

27-Oct-2017: Changes in document version v9.0

No new or updated errata in this document version.

23-Jun-2017: Changes in document version v8.0

ID	Status	Area	Category	Summary
857573	New	Programmer	Category B	Unprivileged load/store operation might incorrectly calculate Watchpoint hit when running in EL2
903758	New	Programmer	Category B (rare)	Load exclusive to pagetable might generate an unexpected external abort

14-Jun-2017: Changes in document version v7.0

ID	Status	Area	Category	Summary
855037	New	Programmer	Category B	MMU might cause incorrect abort checks on changing Exception level
867534	New	Programmer	Category B	Atomic instruction incorrectly triggers a stage2 hardware pagetable update
826172	New	Programmer	Category C	Accessing the CLUSTERTHREADSIDOVR_EL1 register generates an UNDEFINED exception
872696	New	Programmer	Category C	Multiple stores to the same address targeting different types of device memory might cause data corruption
900307	New	Programmer	Category C	Stage2 dirty bit not updated when HCR_EL2.DC set

31-Mar-2017: Changes in document version v6.0

ID	Status	Area	Category	Summary
827291	New	Programmer	Category B	Clearing TCR_ELx.TBI from inside a tagged address region might cause a fault
844522	New	Programmer	Category B (rare)	Advanced SIMD integer multiply instructions in IT block might cause data corruption
846532	New	Programmer	Category B (rare)	Pagetable re-mapping might cause deadlock
827496	New	Programmer	Category C	Data cache maintenance operations by set/way targeting a 4MB L3 cache might affect incorrect cache index

30-Jan-2017: Changes in document version v5.0

ID	Status	Area	Category	Summary
748796	New	Programmer	Category B	Some AT instructions executed from EL3 might incorrectly report a domain fault
784743	New	Programmer	Category B	Aliasing branch instructions might cause an incorrect abort to be taken
785959	New	Programmer	Category B	Store instruction might not cause a stage 2 hardware update
798797	New	Programmer	Category B (rare)	Page table walk to same address as outstanding access might deadlock
765591	New	Programmer	Category C	ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set
777436	New	Programmer	Category C	Speculative hardware pagetable updates might cause an abort when the stage2 pagetables are not cacheable
784215	New	Programmer	Category C	ETM trace reports incorrect branch target
789947	New	Programmer	Category C	DVM might prevent a core power off
795490	New	Programmer	Category C	Incorrect fault status codes used for reporting synchronous uncorrectable ECC aborts
801844	New	Programmer	Category C	ECC error might cause silent data corruption
802429	New	Programmer	Category C	Debug not entering Memory Access mode without setting EDSCR.ERR
804701	New	Programmer	Category C	ESR_EL2.ISV bit set incorrectly for some external aborts
804765	New	Programmer	Category C	Mismatch between EDPRSR.SR and EDPRSR.R

07-Dec-2016: Changes in document version v4.0

ID	Status	Area	Category	Summary
779913	New	Programmer	Category A (rare)	Memory requests from out-of-date page mappings might be observable after a TLB invalidate operation is executed
778703	New	Programmer	Category B	ESB might not contain an error when no L2 cache configured
764819	New	Programmer	Category C	HCR_EL2.DC enabled with VHE affects the PAN behaviour
772155	New	Programmer	Category C	Some PMU events might count incorrectly
773437	New	Programmer	Category C	External abort or uncorrected ECC error might cause further data corruption
776820	New	Programmer	Category C	ESR_ELx.WnR bit set incorrectly for some aborts on atomic instructions

18-Nov-2016: Changes in document version v3.0

ID	Status	Area	Category	Summary
768277	New	Programmer	Category B	Dual-issue of unaligned load or store might not execute correctly
719235	New	Programmer	Category C	Instruction cache parity error might cause an incorrect abort to be taken
764099	New	Programmer	Category C	ECC error on tag RAM combined with external abort or ECC on data RAM might cause deadlock
766365	New	Programmer	Category C	VHE enabled when EL2 is AArch32 causes some memory instructions to incorrectly abort
768143	New	Programmer	Category C	Inconsistent data can be sampled in PMPCSR

26-Oct-2016: Changes in document version v2.0

No new or updated errata in this document version.

06-Oct-2016: Changes in document version v1.0

ID	Status	Area	Category	Summary
752908	New	Programmer	Category C	CTI triggers are captured and remain pending when CTICONTROL.GLBEN is 0
754938	New	Programmer	Category C	ETM might not generate an event packet and ATB trigger
754939	New	Programmer	Category C	ETM might assert AFREADY before all trace has been output
759155	New	Programmer	Category C	Address translation instructions with PAN bit set might incorrectly report an abort in the PAR

Errata summary table

The errata associated with this product affect the product versions described in the following table.

ID	Area	Category	Summary	Found in versions	Fixed in version
779913	Programmer	Category A (rare)	Memory requests from out-of-date page mappings might be observable after a TLB invalidate operation is executed	r0p0	r0p1
748796	Programmer	Category B	Some AT instructions executed from EL3 might incorrectly report a domain fault	r0p0	r0p1
768277	Programmer	Category B	Dual-issue of unaligned load or store might not execute correctly	r0p0	r0p1
778703	Programmer	Category B	ESB might not contain an error when no L2 cache configured	r0p0	r0p1
784743	Programmer	Category B	Aliasing branch instructions might cause an incorrect abort to be taken	r0p0	r0p1
785959	Programmer	Category B	Store instruction might not cause a stage 2 hardware update	r0p0	r0p1
827291	Programmer	Category B	Clearing TCR_ELx.TBI from inside a tagged address region might cause a fault	r0p0, r0p1	r1p0
855037	Programmer	Category B	MMU might cause incorrect abort checks on changing Exception level	r0p0, r0p1	r1p0
857573	Programmer	Category B	Unprivileged load/store operation might incorrectly calculate Watchpoint hit when running in EL2	r0p0, r0p1	r1p0
867534	Programmer	Category B	Atomic instruction incorrectly triggers a stage2 hardware pagetable update	r0p0, r0p1	r1p0
1024718	Programmer	Category B	Update of DBM or AP bits without break-before-make might result in incorrect hardware dirty bit update	r0p0, r0p1, r1p0, r2p0	Open
1530923	Programmer	Category B	Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation	r0p0, r0p1, r1p0, r2p0	Open
798797	Programmer	Category B (rare)	Page table walk to same address as outstanding access might deadlock	r0p0	r0p1
844522	Programmer	Category B (rare)	Advanced SIMD integer multiply instructions in IT block might cause data corruption	r0p0, r0p1	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
846532	Programmer	Category B (rare)	Pagetable re-mapping might cause deadlock	r0p0, r0p1	r1p0
903758	Programmer	Category B (rare)	Load exclusive to pagetable might generate an unexpected external abort	r0p0, r0p1	r1p0
1221012	Programmer	Category B (rare)	A load exclusive following a store or atomic to the same memory location can result in a lack of forward progress	r0p0, r0p1, r1p0	r2p0
2441007	Programmer	Category B (rare)	Completion of affected memory accesses might not be guaranteed by completion of a TLBI	r0p0, r0p1, r1p0, r2p0	Open
719235	Programmer	Category C	Instruction cache parity error might cause an incorrect abort to be taken	r0p0	r0p1
752908	Programmer	Category C	CTI triggers are captured and remain pending when CTICONTROL.GLBEN is 0	r0p0	r0p1
754938	Programmer	Category C	ETM might not generate an event packet and ATB trigger	r0p0	r0p1
754939	Programmer	Category C	ETM might assert AFREADY before all trace has been output	r0p0	r0p1
759155	Programmer	Category C	Address translation instructions with PAN bit set might incorrectly report an abort in the PAR	r0p0	r0p1
764099	Programmer	Category C	ECC error on tag RAM combined with external abort or ECC on data RAM might cause deadlock	r0p0	r0p1
764819	Programmer	Category C	HCR_EL2.DC enabled with VHE affects the PAN behaviour	r0p0	r0p1
765591	Programmer	Category C	ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set	r0p0	r0p1
766365	Programmer	Category C	VHE enabled when EL2 is AArch32 causes some memory instructions to incorrectly abort	r0p0	r0p1
768143	Programmer	Category C	Inconsistent data can be sampled in PMPCSR	r0p0	r0p1
772155	Programmer	Category C	Some PMU events might count incorrectly	r0p0	r0p1
773437	Programmer	Category C	External abort or uncorrected ECC error might cause further data corruption	r0p0	r0p1
776820	Programmer	Category C	ESR_ELx.WnR bit set incorrectly for some aborts on atomic instructions	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
777436	Programmer	Category C	Speculative hardware pagetable updates might cause an abort when the stage2 pagetables are not cacheable	r0p0	r0p1
784215	Programmer	Category C	ETM trace reports incorrect branch target	r0p0	r0p1
789947	Programmer	Category C	DVM might prevent a core power off	r0p0	r0p1
795490	Programmer	Category C	Incorrect fault status codes used for reporting synchronous uncorrectable ECC aborts	r0p0	r0p1
801844	Programmer	Category C	ECC error might cause silent data corruption	r0p0	r0p1
802429	Programmer	Category C	Debug not entering Memory Access mode without setting EDSCR.ERR	r0p0	r0p1
804701	Programmer	Category C	ESR_EL2.ISV bit set incorrectly for some external aborts	r0p0	r0p1
804765	Programmer	Category C	Mismatch between EDPRSR.SR and EDPRSR.R	r0p0	r0p1
826172	Programmer	Category C	Accessing the CLUSTERTHREADSIDOVR_EL1 register generates an UNDEFINED exception	r0p0, r0p1	r1p0
827496	Programmer	Category C	Data cache maintenance operations by set/way targeting a 4MB L3 cache might affect incorrect cache index	r0p0, r0p1	r1p0
872696	Programmer	Category C	Multiple stores to the same address targeting different types of device memory might cause data corruption	r0p0, r0p1	r1p0
900307	Programmer	Category C	Stage2 dirty bit not updated when HCR_EL2.DC set	r0p0, r0p1	r1p0
948532	Programmer	Category C	Incorrect PMCEID1 value	r0p0, r0p1, r1p0	r2p0
999993	Programmer	Category C	Multiple concurrent ECC errors might cause silent data corruption	r0p0, r0p1, r1p0, r2p0	Open
1017312	Programmer	Category C	DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might be erroneous	r0p0, r0p1, r1p0	r2p0
1030596	Programmer	Category C	Cycle count value in timestamp packet might be incorrect	r0p0, r0p1, r1p0, r2p0	Open
1030597	Programmer	Category C	Incorrect ETM timestamp value when timestamp and event generation happen on same cycle	r0p0, r0p1, r1p0, r2p0	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
1131793	Programmer	Category C	Stores might prevent forward progress of an exclusive loop	r0p0, r0p1, r1p0, r2p0	Open
1214720	Programmer	Category C	VFP_SPEC PMU event might count incorrectly	r0p0, r0p1, r1p0, r2p0	Open
1401736	Programmer	Category C	Non-cacheable loads of mismatched size might not be single-copy atomic	r0p0, r0p1, r1p0, r2p0	Open
2109978	Programmer	Category C	Atomic store instructions might not report an External abort or System Error	r0p0, r0p1, r1p0, r2p0	Open
2288055	Programmer	Category C	DTR flags not cleared on external debugger access while leaving Debug state	r0p0, r0p1, r1p0, r2p0	Open
2342475	Programmer	Category C	Halting step syndrome might be wrong on stepping a Load-Exclusive instruction	r0p0, r0p1, r1p0, r2p0	Open
2364958	Programmer	Category C	Some unallocated debug and trace System registers might be trapped by HCR_EL2.TIDCP	r0p0, r0p1, r1p0, r2p0	Open

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

779913
Memory requests from out-of-date page mappings might be observable after a TLB invalidate operation is executed

Status

Affects: Cortex-A55
Fault Type: Programmer Category A Rare
Fault Status: Present in r0p0. Fixed in r0p1.

Description

A TLB invalidate operation is defined to be complete only when all memory accesses using the TLB entries that have been invalidated have been observed by all observers if those accesses are required to be observed. However, if a TLB invalidate operation is executed on one core at the same time as a load or store instruction to the virtual address being invalidated is executed on another core, then because of this erratum the TLB operation might complete before all the load or store instruction has completed.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. Core A executes a load, store, or atomic instruction. This could be an LDM, STM, VLDM, VSTM, LDP, or LDNP instruction that accesses at least two doublewords of memory, or a v8.1 atomic instruction.
2. Core B writes to the page tables to modify the translation for the virtual address used by the instruction executing on Core A. Note that because of TLB caching, this write might happen before or during the instruction being executed on Core A.
3. Core B executes an Inner Shareable TLB maintenance instruction that invalidates the virtual address of the load or store being executed on core A.
4. The TLB invalidate instruction is executed at the same time as the load or store is being executed on core A.
5. Core B executes a DSB instruction. There might be other instructions executed between the TLB invalidate and the DSB.

Implications

This erratum could cause part of the load, store, or atomic instruction to happen using an old address translation after the operating system thinks that all accesses using that old translation have completed. All parts of the instruction that are to the same 64byte address will use the same translation, and the DSB will not complete until at least one part of the load, store, or atomic instruction has completed.
In the significant majority of the cases when this erratum occurs, especially when accessing cacheable memory, the load, store, or atomic instruction executing will have completed naturally before the TLB invalidate and DSB complete, and so the effect of this erratum would not be visible to software.
Note that this erratum has been designated rare as it has been assessed by ARM to be acceptable for engineering samples.

Workaround

There is no viable workaround. Given that operating systems will have been constructed assuming that these TLB invalidations affect everything in inner shareable domain, it is not viable to modify them to use a different approach.

Category B

748796
Some AT instructions executed from EL3 might incorrectly report a domain fault

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Address translation instructions executed from EL3 and targeting EL1 or EL0 might report an incorrect result in the PAR when the HCR_EL2.DC bit is set.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The core is executing at Exception level 3 in AArch64.
2. The core executes one of the following address translation instructions:
   - ATS1E01
   - ATS1E1P
   - ATS1E01P
3. The Exception level targeted by the address translation instruction is Non-secure and AArch32.
4. HCR_EL2.DC is set to 1.
5. HCR_EL2.E2H is set to 0.
6. The DACR is programmed so that domain 0 would cause a domain fault if the HCR_EL2.DC bit had not been set. Note that this is the default value out of reset.

Implications

The PAR register will incorrectly report that a domain fault occurred.

Workaround

Secure software can set the DACR[1:0] to 0b01 before executing the address translation instruction. It should restore the previous DACR value before returning to a lower Exception level.

768277
Dual-issue of unaligned load or store might not execute correctly

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor supports dual-issuing of a load and a store instruction together to improve performance. When one or both of the load and store instructions are unaligned, then if some additional conditions also occur at the same time it can result in incorrect execution of the instructions.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. A load instruction and a store instruction are executed. This could be in either order.
2. No other instructions are executed between the load and the store instruction.
3. One or both of the instructions are to an unaligned address.
4. Various other microarchitectural conditions occur in addition to the above sequence.

Implications

Depending on the other conditions that occur, this erratum might result in deadlock, data corruption, taking an incorrect abort, or speculatively accessing a memory location that should not have been accessed.

Workaround

The erratum can be avoided by disabling dual-issue of load instructions with store instructions. This will have a small impact on performance. To do this, set CPUACTLR(_EL1)[31] to 1.

778703
ESB might not contain an error when no L2 cache is configured

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

When a partial cache line write that does not allocate in the L1 receives an external abort, data corruption that is not fully contained by an ESB instruction might occur.

Configurations Affected

This erratum only affects configurations of the Cortex-A55 processor where L2_CACHE is false.

Conditions

1. There is a store that does not merge into a full cache line in the store buffer and does not allocate in the L1.
2. The store receives an external abort from the interconnect.
3. Another core issues an access to the same cache line around the same time as the store.
4. The second access does not receive an external abort from the interconnect.

Implications

If the erratum occurs, there will be data corruption in the cache line, and an imprecise asynchronous abort correctly reported to indicate a UEU error. However, there might be further data corruption to the same cache line after the abort is reported, and so an ESB instruction will not be sufficient to contain the error. Therefore if that physical address is reused after the error has been handled, then the data corruption could affect the new use of that address. Therefore the result is a small increase in the failure in time (FIT) rate.

Workaround

For designs where RAS is of significant importance, this erratum can be worked around by writing 0b11 to CPUECTLR(_EL1)[26:25] to disable write-streaming mode and writing 1b1 to CPUACTRL[24] to disable write-streaming of transient and non-temporal stores. This will have an impact on performance for streaming workloads. Note that this should only be done when there is no private L2 cache configured, which is indicated by CLIDR(_EL1)[8:6] reading 0b000.

784743
Aliasing branches might cause an incorrect abort to be taken

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor contains various program flow prediction hardware structures, and if two branches with the same virtual address alias to the same entry in one of the prediction structures, then under some conditions it is possible for a fault to be taken when no instruction from the faulting page was architecturally executed.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The program counter points to a 16-bit length branch instruction which is at the end of a page. The page which is immediately after the branch instruction would cause a translation fault, access flag fault, or permission fault if it was accessed.
3. There was another branch instruction at the same virtual address in the past, that was a 32-bit instruction. This could have been because of self-modifying code, or from a different process running under a different ASID, VMID, or Exception level.
4. The target of both branches is also the same virtual address.

Implications

If the above conditions are met, then the translation, permission, or access flag fault is incorrectly taken on the target of the 16-bit branch instruction that is in a non-faulting page. Some operating systems may already be tolerant to this behavior.

Workaround

The abort handling software can check that the address in the FAR_ELx matches an address that is expected to abort. If the address was not expected to abort, then the abort handler can return and re-execute the instruction. Because the abort is caused by the branch, but is only taken on the target of the branch, when the target of the branch is re-executed it will not abort for the same reason.
If there is no abort handling software that is expected to deal with such aborts, for example in a hypervisor or secure firmware, then the system might not be tolerant to an unexpected abort. In such cases, the software must ensure that the following page after the end of the instruction code is accessible and executable.

785959
Store instruction might not cause a stage 2 hardware update

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a hypervisor enables the stage 2 hardware dirty state update, then when both the Privileged Access Never and Default Cacheable control bits are set, a privileged store instruction might execute without causing a stage 2 hardware dirty state update.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. EL2 is AArch64.
2. VTCR_EL2.HA and VTCR_EL2.HD are both set.
3. HCR_EL2.DC is set.
4. PSTATE.PAN is set.
5. A privileged store instruction is executed at Non-secure EL1.
6. The stage 2 page table descriptor used by the store has the DBM bit set and S2AP[1] clear, indicating no stage 2 write permission.

Implications

If the erratum occurs, the store instruction will complete, but the page table descriptor will not be updated. This can cause the page to have been dirtied without the hypervisor being aware of it, which could lead to the loss of that data if the hypervisor later reuses that page without realizing it needs to be paged out.

Workaround

The hypervisor should not enable hardware dirty bit updates on stage 2 page tables, and should instead use software handling of the dirty status of the page.

827291
Clearing TCR_ELx.TBI from inside a tagged address region might cause a fault

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r1p0.

Description

The ARMv8 architecture supports address tagging in the AArch64 execution state. When address tagging is used, the top byte of the virtual address is ignored during address translation. If a branch is executed, tag bits should be removed from the target address when the program counter is updated. Because of this erratum, the tag bits might propagate to the program counter. If the Top Byte Ignored bit is subsequently cleared in the relevant Translation Control Register, then the processor might report a fault because the program counter is out of range.

Configurations Affected

All configurations are affected.

Conditions

1. The TBI bit in the Translation Control Register for the current Exception level is set to 1.
2. The core executes an indirect branch to an address where the top byte is not sign-extended.
3. The branch was incorrectly predicted not-taken before its execution.
4. The TBI bit is set to 0 by the code in the address-tagged region.
5. A context synchronization event occurs.

Implications

If the above conditions are met, then the address tag might propagate to an internal copy of the program counter. This can cause an unexpected translation fault or address size fault after the context synchronization event. Reads of the PC register are not affected.

Workaround

The erratum can be avoided by inserting an ISB instruction before the instruction that clears the TBI bit.

855037
MMU might cause incorrect abort checks on changing Exception level

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r1p0.

Description

The MMU on status is used to calculate stage 1 permission and domain faults. If the MMU on status changes because of an Exception level change, then under certain conditions the incorrect MMU on status can be used for an instruction fetch abort check.

Configurations Affected

All configurations are affected.

Conditions

1. The processor is executing in AArch32 state, using the short pagetable descriptor format.
2. An exception or exception return is taken to a different Exception level. The TrustZone security state must not change.
3. An address translation completes for an instruction fetch with specific microarchitectural timing relative to the Exception level change.
4. The effective value of the relevant SCTLR_ELx.M bit, after modification by the HCR_EL2.TGE/DC bits, has a different value in the new Exception level compared to the old Exception level.
5. Execution returns to the previous Exception level.
6. Between conditions 2 and 5, there are no TLB invalidate instructions executed, and no system register writes to any of the following registers: SCR, SCTLR, HSCTLR, SCTLR_ELx, HCR, HCR2, HCR_EL2, and DACR.

Implications

If the above conditions are met, the stage 1 permission checking and domain fault checking logic will use an incorrect value for the MMU on status for the translation performed in condition 3. This can cause a Prefetch Abort to be incorrectly taken, or no Prefetch Abort to be taken when it should.

Workaround

Software using AArch64 or the long pagetable descriptor format does not need a workaround. For AArch32 software using the short pagetable descriptor format, there is no workaround.

857573
Unprivileged load/store operation might incorrectly calculate Watchpoint hit when running in EL2

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

When the processor is running AArch64 code in EL2, and the Virtualization Host Extensions are enabled, then a Load/Store Unprivileged instruction might incorrectly calculate whether a Watchpoint hits.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The processor is executing AArch64 code at EL2.
2. HCR_EL2.E2H and HCR_EL2.TGE are both set.
3. DBGWCR<n>_EL1.{HMC, SSC, PAC} are set to {0b1,0b11,0b00}, {0b0,0b00,0b11}, or {0b0,0b01,0b11}
4. A Load/Store Unprivileged instruction is executed, which matches all other Watchpoint conditions other than the HMC, SSC, and PAC bits.

Implications

If the above conditions are met, the load/store Unprivileged instruction will incorrectly calculate a Watchpoint hit. If the {HMC, SSC, PAC} bits are set to {0b1,0b11,0b00} then it will generate a Watchpoint exception when it should not, otherwise it will not generate a Watchpoint exception when it should.

Workaround

For unexpected watchpoints, the debugger can check the Exception level and type of instruction that triggered the watchpoint and if it was a load/store Unprivileged that was not expected to match the watchpoint then the debugger should restart execution of the program.

867534
Atomic instruction incorrectly triggers a stage2 hardware pagetable update

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

When stage2 pagetable hardware updates of the dirty bit are enabled, and a page is marked as no access, then executing an atomic instruction can incorrectly cause the dirty bit to be set.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The processor is executing in AArch64 state at Non-secure EL0 or EL1.
2. The VTCR_EL2.HA and VTCR_EL2.HD bits are both set.
3. Stage2 translations are enabled (either by setting HCR_EL2.VM or HCR_EL2.DC).
4. An atomic instruction is executed.
5. The stage2 page/block descriptor that matches the virtual address of the atomic instruction has DBM=1 and S2AP=0b00.

Implications

If the combined stage1 and stage2 memory type is not write-back, then an atomic update fault will be incorrectly taken instead of a permission fault. Otherwise, the page/block descriptor S2AP field will be incorrectly updated to 0b10 (write only) in addition to taking the permission fault.

Workaround

If the hypervisor uses no access permissions at stage2, then it should not enable hardware dirty bit updates on stage 2 page tables, and should instead use software handling of the dirty status of the page.

1024718
Update of DBM or AP bits without break-before-make might result in incorrect hardware dirty bit update

Status

Affects: Cortex-A55
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

If the hardware dirty bit update is enabled, and the DBM or AP bits in the translation table descriptor are updated by software without using break-before-make, then it is possible for hardware to incorrectly update the AP bits based on the old value of those bits.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The hardware dirty bit update is enabled for stage1 (TCR_ELx.HA and TCR_ELx.HD are both set) or stage2 (VTCR_EL2.HA and VTCR_EL2.HD are both set).
2. A store instruction is executed, which causes a hardware dirty bit update for the translation table descriptor which has DBM=1 and no write permission because of either AP[2]=1 or S2AP[1]=0.
3. At the same time as the store, the OS or hypervisor writes to the same translation table descriptor to update the AP, S2AP, or DBM bits, without using a break-before-make procedure.
4. The new translation table descriptor is a valid translation, but does not require the dirty bit update for the store because either the DBM bit is now clear, or the AP/S2AP bits would still cause a permission fault to occur even after a dirty bit update.

Implications

The permission checking of the store is performed on the old version of the translation table descriptor, while the AP/S2AP bit is updated in the new version of the translation table descriptor. This could lead to an inconsistent situation where the store is executed but the OS or hypervisor is not expecting the store to have permission to execute.

Workaround

The OS or hypervisor can use a break-before-make procedure if it needs to update the DBM or AP/S2AP bits. Alternatively, it can use software management of the dirty bit update.

1530923
Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

A speculative Address Translation (AT) instruction translates using registers that are associated with an out-of-context translation regime and caches the resulting translation in the TLB. A subsequent translation request that is generated when the out-of-context translation regime is current uses the previous cached TLB entry producing an incorrect virtual to physical mapping.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A speculative AT instruction performs a table walk, translating a virtual address to a physical address using registers associated with an out-of-context translation regime.
2. Address translation data that is generated during the walk is cached in the TLB.
3. The out-of-context translation regime becomes current and a subsequent memory access is translated using previously cached address translation data in the TLB, resulting in an incorrect virtual to physical mapping.

Implications

If the above conditions are met, the resulting translation would be incorrect.

Workaround

When context-switching the register state for an out-of-context translation regime, system software at EL2 or above must ensure that all intermediate states during the context-switch would report a level 0 translation fault in response to an AT instruction targeting the out-of-context translation regime. A workaround is only required if the system software contains an AT instruction as part of an executable page.

Category B (rare)

798797
Page table walk to same address as outstanding access might deadlock

Status

Affects: Cortex-A55
Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a pagewalk occurs at the same time as one or more loads or stores to the pagetable entries in the same cache line as the pagewalk is accessing, then under certain additional timing conditions the processor might deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. A linefill is started to the L1 data cache for virtual address VA1 and physical address PA1. This could be started by a load, store, or the prefetch.
2. Bits VA1[13:12] need to be different to PA1[13:12].
3. The TLB makes a second linefill request for PA1. The TLB linefill request must happen while the other linefill is still in progress.
4. Additional complex microarchitectural timing conditions occur.

Implications

It is expected that it would be very difficult for typical software to manage to trigger all of these conditions. If the conditions are triggered then the processor might deadlock.

Workaround

The erratum can be avoided by preventing pagewalks from allocating lines into the L1 cache. This will have a negligible impact on performance when an L2 cache is present. To do this, set CPUACTLR(_EL1)[49] to 1.

844522
Advanced SIMD integer multiply instructions in IT block might cause data corruption

Status

Affects: Cortex-A55
Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

The Cortex-A55 processor can optionally be configured to support Advanced SIMD instructions.

The Cortex-A55 processor also supports conditional execution of instructions. In the T32 execution state, when the processor executes an `IT` instruction, up to four instructions at a time can become conditional.

Because of this erratum, if certain Advanced SIMD integer multiply instructions are executed conditionally, the processor might produce the wrong result for subsequent and dependent Advanced SIMD integer multiply-accumulate instructions.

Configurations affected

The processor must be configured with Advanced SIMD support.

Conditions

This erratum occurs when the processor executes a sequence of instructions in the T32 execution state. The sequence must comprise of the following instructions:

1. An `IT` instruction.
2. An optional instruction, belonging in the IT block.
3. An Advanced SIMD producer instruction, belonging in the IT block.
4. An optional instruction, belonging in the IT block.
5. An Advanced SIMD consumer instruction, which may or may not belong in the IT block.

Additionally, all the following conditions must hold:

- The producer instruction (3) must be one of the following:
  - VMLA (by scalar).
  - VMLA (integer).
  - VMLAL (by scalar).
  - VMLAL (integer).
  - VMLS (by scalar).

    - VMLS (integer).
    - VMLSL (by scalar).
    - VMLSL (integer).
    - VMUL (by scalar).
    - VMUL (integer).
    - VMULL (by scalar).
    - VMULL (integer).
- The consumer instruction (5) must be one of the following:
    - VMLA (by scalar).
    - VMLA (integer).
    - VMLAL (by scalar).
    - VMLAL (integer).
    - VMLS (by scalar).
    - VMLS (integer).
    - VMLSL (by scalar).
    - VMLSL (integer).
- The <Qd> or <Dd> destination operand of the producer instruction must match the <Qd> or <Dd> source operand of the consumer instruction.
- The <dt> data type of the producer instruction must be compatible with the <dt> data type of the consumer instruction.
- Both the producer instruction and the consumer instructions must be of the integer variant, and not of the floating-point or the polynomial variants.
- The producer instruction must fail its condition code check.
- The consumer instruction must either:
    - Not belong in the IT block and therefore execute unconditionally.
    - Belong in the IT block and pass its condition code check, because either:
        - The consumer has the opposite condition code compared to the producer.
        - The consumer has the same condition code compared to the producer, but a flag-setting instruction executes successfully in (4) and changes the condition flags.

Even though all the conditions above might hold, the erratum does not always occur. Specifically, any of the following conditions might affect its occurrence:

- The timing of the instructions in the processor pipeline.
- The arrangement of the instructions in dual-issuing pairs.

NOTE

Note that the ARM architecture deprecates the conditional execution of any instruction encoding provided by Advanced SIMD that is not also provided by floating-point. The architecture strongly recommends that such T32 Advanced SIMD instructions are never included in an IT block.

Implications

If this erratum occurs, the processor will incorrectly forward parts of the resulting value of the <Qd> or <Dd> operand from the producer instruction to the consumer instruction, even though the producer instruction has failed its condition code check. Therefore, the result of the consumer instruction will be wrong.

Workaround

A workaround is not expected to be required in most cases. This is because the erratum conditions require use of deprecated code.
If the erratum is encountered, it is preferable to work around it by avoiding the conditions described. If it is not possible to modify the code then contact ARM support for more details.

846532
Pagetable remapping might cause deadlock

Status

Affects: Cortex-A55
Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

When the pagetables are reprogrammed to change a translation, and a specific code sequence that uses the same translation is executed during the reprogramming, the processor might deadlock while performing a load instruction.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. There is a valid translation for virtual address A, which maps to physical address B and is marked as Normal Cacheable memory.
2. The pagetables are altered so that either:
   - The translation maps to a different physical address.
   - The translation would cause a fault, but the bits of the pagetable entry that would correspond to the physical address if it were a valid translation are different to physical address B.
3. A load and store instruction are executed back to back, in any order.
4. The load address is not aligned to its access size.
5. The load and store instructions both have virtual addresses within the same 4KB memory region as virtual address A.
6. A further load instruction is executed addressing the same 4KB memory region.
7. The original translation is naturally evicted from the TLB by a prefetch or speculative access, between the first load and the store performing their translations.
8. Both loads and the store execute before a TLB invalidate instruction forces the original translation to be removed from the TLB.

If these conditions are met under certain timing conditions, the final load instruction might not complete, which would cause the processor to deadlock.

Implications

In rare circumstances, the processor could deadlock.

Workaround

Given the rarity of the conditions needed to trigger this erratum, a workaround is not expected to be needed in most systems. If a workaround is required, then one of the following options can be used:

- The erratum can be avoided by disabling dual-issue of load instructions with store instructions. This will have a small impact on performance. To do this, set CPUACTLR(_EL1)[31] to 1.
- An alternate workaround is to disable a power optimization feature. This will have no impact on performance, but will increase power consumption. To do this, set CPUACTLR(_EL1)[34] to 1.

903758
Load exclusive to page table might generate an unexpected external abort

Status

Affects: Cortex-A55
Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

If the MMU performs a pagewalk at the same time as a load exclusive to the same address, then the access that started the pagewalk can incorrectly receive a synchronous external abort.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The page table descriptors are stored in cacheable memory.
2. A load exclusive instruction is executed.
3. The load exclusive is to a physical address that is used for the page tables. This address misses in the L1 data cache.
4. The MMU starts a pagewalk. This could be because of an instruction fetch or a speculative data access.
5. The pagewalk reads the same cache line as the load exclusive instruction. This must happen before the linefill started by the load exclusive has completed.

Implications

If the above conditions are met, the access that started the pagewalk might complete incorrectly with a synchronous external abort on the translation table walk. Note that the pagewalk must be started by the same core that started the exclusive access. This implies that the same core must be both updating the page table and making use of that translation at the same time.

Workaround

The erratum can be avoided by preventing pagewalks from allocating lines into the L1 cache. This will have a negligible impact on performance when an L2 cache is present. To do this, set CPUACTLR(_EL1) [49] to 1.

1221012
A load exclusive following a store or atomic to the same memory location might result in a lack of progress

Status

Affects: Cortex-A55
Fault Type: Programmer Category B Rare
Fault Status: Present in r0p0, r0p1, and r1p0. Fixed in r2p0.

Description

If a Cortex-A55 core executes a load exclusive instruction after a non-exclusive store or atomic to the same cache line, and the store meets certain conditions, then a deadlock might occur if the same cache line is being written to, or accessed in a successful exclusive sequence, on one or more other cores in the system.

Configurations affected

This erratum affects all configurations.

Conditions

1. The core executes a non-exclusive store or atomic.
2. The store or atomic is a store release or atomic instruction, a non-temporal store (either by instruction type or by memory attributes using the transient bit in the translation tables), or any other type of store to a virtual alias of a later load exclusive.
3. The core executes other unrelated stores or cache maintenance operations.
4. The core executes a load exclusive to the same cache line as the initial non-exclusive store or atomic.
5. The load, store, or atomic instructions from steps 1 and 4 target the same cache line and all have memory attributes that are Inner Write Back Outer Write Back cacheable.
6. One or more cores in the system are executing relatively continuous store instructions to the same cache line.

Implications

If this erratum occurs, then the core does not make progress until other cores in the system stop executing store instructions to the same cache line.

Workaround

Software can work around this erratum by inserting a DMB.LD before any load exclusive on a Cortex-A55 core. This can be performed automatically by hardware if the following code sequence is executed from EL3 once at boot time:

AArch64 sequence

```
mov  x0, #0x0020
movk x0, #0x0850, lsl #16
msr  s3_6_c15_c8_2, x0

mov  x0, #0x0000
movk x0, #0x1FF0, lsl #16
movk x0, #0x2, lsl #32
msr  s3_6_c15_c8_3, x0

mov  x0, #0x03fd
movk x0, #0x0110, lsl #16
msr  s3_6_c15_c8_1, x0

mov  x0, #0x1
msr  s3_6_c15_c8_0, x0

mov  x0, #0x0040
movk x0, #0x08D0, lsl #16
msr  s3_6_c15_c8_2, x0

mov  x0, #0x0040
movk x0, #0x1FF0, lsl #16
movk x0, #0x2, lsl #32
msr  s3_6_c15_c8_3, x0

mov  x0, #0x03fd
movk x0, #0x0110, lsl #16
msr  s3_6_c15_c8_1, x0
```

AArch32 sequence

```
mov  r0, #0x0000
movt r0, #0x0850
mov  r1, #0
mcrr p15, 9, r0, r1, c15

mov  r0, #0x0000
movt r0, #0x1FF0
mov  r1, #0x2
mcrr p15, 10, r0, r1, c15

mov  r0, #0x03FD
movt r0, #0x0110
mov  r1, #0x0
mcrr p15, 8, r0, r1, c15

mov  r0, #0x1
mcr  p15, 6, r0, c15, c8, 0
mov  r0, #0x0040
movt r0, #0x08D0
mov  r1, #0
mcrr p15, 9, r0, r1, c15

mov  r0, #0x0040
movt r0, #0x1FF0
mov  r1, #0
mcrr p15, 10, r0, r1, c15

mov  r0, #0x3FD
movt r0, #0x0110
mov  r1, #0x0
mcrr p15, 8, r0, r1, c15
```

2441007
Completion of affected memory accesses might not be guaranteed by completion of a TLBI

Status

Affects: Cortex-A55
Fault Type: Programmer Category B (Rare)
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

The core might not guarantee completion of all memory accesses after completion of a TLB Invalidate (`TLBI`) instruction affecting those accesses on another core.

Configurations affected

This erratum affects all configurations.

Conditions

1. Another PE in the system executes a `TLBI` or Instruction Cache (`IC`) instruction, followed by a Data Synchronization Barrier (`DSB`) instruction.
2. The core executes a store to a memory location A.
3. Another PE in the system modifies the descriptor used by the store to memory location A, using a break-before-make sequence.
    - The break-before-make sequence will include a `TLBI` instruction, followed by a `DSB` instruction.
4. Rare, timing-sensitive, microarchitectural conditions occur.

Implications

The `DSB` used after the `TLBI` as part of the break-before-make sequence might not guarantee the completion of the store to memory location A under very rare and unlikely timing conditions. For most systems and applications, the latency of the break-before-make sequence and time until later reuse is very likely to exceed the latency required to naturally complete the store.

Workaround

Given the rarity of the conditions needed to trigger this erratum, a workaround is not expected to be needed in most systems.
If a workaround is required, then the `TLBI`, `DSB` sequence from the break-before-make sequence can be repeated. After repeating the `TLBI`, `DSB` sequence, all memory accesses that use a translation changed by the break-before-make sequence will have completed.

Category C

719235
Instruction cache parity error might cause an incorrect abort to be taken

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The core cache protection option allows the Cortex-A55 processor to detect and correct a 1-bit error in the instruction cache RAMs. If an error is detected on an instruction at the end of a page, and the following page would generate a fault if executed, then under some conditions it is possible for a fault to be taken when no instruction from the faulting page was architecturally executed.

Configurations affected

This erratum affects configurations of the Cortex-A55 processor with CORE_CACHE_PROTECTION enabled.

Conditions

1. The core is executing in AArch32 T32 instruction state.
2. The program counter points to a 16-bit length instruction which is at the end of a page.
3. The related instruction is stored in the L1 instruction cache.
4. There is a single bit error on one specific bit of the instruction, which causes the instruction to be incorrectly interpreted as a 32-bit instruction.
5. The page which is immediately after the address range of the previous 16-bit instruction would cause a translation fault, access flag fault, or permission fault if it was accessed.

Implications

If the above conditions are met, then the translation, permission, or access flag fault is incorrectly taken, instead of correcting the error. There is still substantial benefit being gained from the parity logic. There might be a negligible increase in overall system failure rate due to this erratum.

Workaround

No workaround is required.

752908
CTI triggers are captured and remain pending when CTICONTROL.GLBEN is 0

Status

Affects: Cortex-A55
Fault type: Programmer Category C
Fault status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor integrates an embedded cross-trigger interface with each core. Because of this erratum, events which occur while the CTI is disabled and should be ignored can cause triggers once the CTI is enabled.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The CTI is disabled, with CTICONTROL.GLBEN = 0.
2. An event occurs which would generate an input trigger to the CTI.
3. The CTI is subsequently enabled by setting CTICONTROL.GLBEN to 1.

Implications

If the conditions are met, then after the CTI is enabled in step 3 the CTI might receive spurious input triggers.

Workaround

There is no workaround for this issue.

754938
ETM might not generate an event packet and ATB trigger

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

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 erratum, 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-A55 processor.

Conditions

1. The ETM is becoming idle because of any of the following:
   - The ETM has been disabled because TRCPRGCTLR.EN = 0 or TRCOSLAR.OSLK = 1.
   - The core has started to execute a WFI or WFE and is going to enter sleep mode.
   - **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 erratum will only be noticed when trying to correlate the CTI behavior with the trace stream.

Workaround

There is no workaround for this erratum.

754939
ETM might assert AFREADY before all trace has been output

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

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-A55 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.

759155
Address translation instructions with PAN bit set might incorrectly report an abort in the PAR

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Address translation instructions that execute from EL3 or EL2 using an EL1 translation regime when EL1 is using AArch32, might incorrectly report an abort in the PAR if the PAN bit is set and the access is to a Manager domain.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The processor is executing in AArch64 state at EL2 or EL3.
2. The processor executes an AT S1E1RP or AT S1E1WP instruction.
3. The EL1 context is using AArch32 short-descriptor format page tables.
4. The DACR is set so that the address being translated is mapped as a Manager domain.
5. The PSTATE.PAN bit is set, and the page table is set up so that the access would have aborted if the domain was set to Client.

Implications

Software using address translation instructions under these conditions might see an abort reported in the PAR when a load or store executed in the same translation regime would not cause an abort.

Workaround

If software is not using a mixture of Manager domains and the PAN bit, then no workaround is necessary. If the EL2/EL3 software requires the translated address in these conditions, then it can use the AT S1E1R or AT S1E1W instead.

764099
ECC error on tag RAM combined with external abort or ECC on data RAM might cause deadlock

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a load instruction gets data returned with an external abort or poisoned data because of an uncorrectable ECC error, then when further conditions occur including another ECC error, the core might deadlock.

Configurations Affected

This erratum affects configurations of the Cortex-A55 processor with CORE_CACHE_PROTECTION enabled.

Conditions

1. A store instruction is executed to writeback cacheable memory.
2. The store is either to an address marked in the pagetables as transient or no write allocate, or the store is non-temporal or has triggered write streaming mode.
3. A load address to the same cache line is executed, looks up in the cache, and misses.
4. The load data is returned from L2, L3, or the external system, with either poisoned data or an external abort.
5. The load looks up in the cache a second time, because of an address hazard against the store.
6. The load gets an ECC error on a tag during the second lookup.

Implications

The result is a small increase in detected uncorrected error (DUE) failure in time (FIT) rate.
Note that in addition to the other conditions, this erratum requires an external abort, or an uncorrectable ECC error on the data, both of which are already likely to be fatal to the process running.

Workaround

No workaround is required.

764819
HCR_EL2.DC enabled with VHE affects the PAN behavior

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If the Virtualization Host Extensions (VHE) are in use with an unnecessary configuration of the HCR_EL2.DC bit, then the Privileged Access Never (PAN) mechanism is not applied correctly.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The core is executing code at EL2 in AArch64 state.
2. The HCR_EL2.E2H bit is set to 1.
3. The HCR_EL2.TGE bit is set to 1.
4. The PSTATE.PAN bit is set to 1.
5. The HCR_EL2.DC bit is set to 1.
6. A load or store instruction is executed, with an address that has access permitted at EL0.

Implications

The HCR_EL2.DC bit is defined by the architecture to have no effect when both the E2H and TGE bits are set. Therefore, this erratum can only occur if EL2 software unnecessarily sets the DC bit. If these conditions are met, then the load or store will not take an abort, as the effect of the PAN bit is not correctly taken into account.

Workaround

For most software, no workaround is expected to be needed. Software running with E2H and TGE bits set should clear the DC bit.

765591
ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

An ATS12NSOPR address translation instruction executed from EL3 might report an incorrect result in the PAR when both the HCR.TGE bit and the SCTLR.M bit are set.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The core is executing at Exception level 3 in AArch32.
2. SCR.NS = 0.
3. HCR.TGE = 1.
4. SCTLR(ns).M = 1.
5. The core executes an ATS12NSOPR instruction.

Implications

The PAR register will incorrectly report the translation as if the stage 1 MMU was enabled.
Note that the combination of both HCR.TGE and SCTLR.M bits being set was UNPREDICTABLE in earlier versions of the architecture, and was only given a defined behavior to reduce the UNPREDICTABLE space. It is not expected to be a useful combination for software.

Workaround

Secure software can clear the SCTLR(NS).M bit before executing the address translation instruction, if the HCR.TGE bit is set. It should restore the previous SCTLR(NS).M value before returning to a lower Exception level.

766365
VHE enabled when EL2 is AArch32 causes some memory instructions to incorrectly abort

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Virtualization Host Extensions (VHE) are an addition in the v8.1 architecture for AArch64 state. If VHE is enabled when running in AArch32, then some load and store instructions might incorrectly cause a Data Abort.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. Exception level 3 is using AArch64 state.
2. SCR_EL3.RW is set to 0, so that EL2 and below are using AArch32 state.
3. HCR.TGE is set to 1.
4. The RES0 bit HCR2[2] is set to 1. This bit corresponds to HCR_EL2.E2H when in AArch64.
5. A load, store, or other memory instruction that requires an address translation is executed.
6. Either the current Exception level is EL0, or the instruction requires the translation to be treated as non-privileged.
7. Depending on the values in some of the fields in the pagetable entry, the access might incorrectly report a Data Abort.

Implications

VHE is only supported when EL2 is AArch64, so there is no need to set the HCR2[2] bit when EL2 is AArch32. Therefore, this erratum can only occur if EL2 software unnecessarily sets the bit, or if EL3 does not initialize or clear the bit when switching EL2 from AArch64 to AArch32.

Workaround

For most software, no workaround is expected to be needed. If software running at EL3 is switching EL2 between AArch64 and AArch32, then it should ensure that it clears the HCR2[2] bit before returning to EL2.

768143
Inconsistent data can be sampled in PMPCSR

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor implements the v8.2 PC Sample-based Profiling extension. A read of PMPCSR[31:0] will return the low 32 bits of the PC of a recently executed instruction, and write several other registers with additional information about the sampled instruction. Because of this erratum, the PMPCSR.NS and PMPCSR.EL fields might contain incorrect information.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The processor executes an instruction or takes an interrupt which causes a change in Exception level or security state.
2. Before the first instruction at the new exception level and security state is executed, PMPCSR[31:0] is read.
3. Subsequently, PMPCSR[63:32] is read.

If these conditions are met, then the read PMPCSR.PC value will reflect the PC of the last instruction executed before the change in Exception level or security state, but the read PMPCSR.NS and PMPCSR.EL values will report the new Exception level and security state, not the state at the time the instruction was executed.

Implications

When performing PC Sample-based profiling there will be minor inaccuracies in the NS and EL fields in the sampled data. This erratum does not affect whether or not sampling will be prohibited, and so a Secure PC value will never be sampled if Secure non-invasive debug is not permitted.

Workaround

There is no workaround for this erratum.

772155
Some PMU events might count incorrectly

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor implements several architectural and implementation-defined performance monitor events. Because of this erratum, some of the events will not count correctly.

Configurations Affected

All configurations are affected.

Conditions

1. One of the PMU event counters is configured to count one of the following events:
    - 0x000F, UNALIGNED_LDST_RETIRED
    - 0x0038, REMOTE_ACCESS_RD
    - 0x0050, L2D_CACHE_RD
    - 0x0051, L2D_CACHE_WR
    - 0x0052, L2D_CACHE_REFILL_RD
    - 0x0053, L2D_CACHE_REFILL_WR
    - 0x074, ASE_SPEC
    - 0x075, VFP_SPEC
    - 0x076, PC_WRITE_SPEC
    - 0x077, CRYPTO_SPEC
    - 0x0C7, L3D_WS_MODE
    - 0x00EC, STALL_BACKEND_ST_TLB

Implications

Software will not be able to use the above events for performance analysis.

Workaround

There is no workaround for this erratum. In some situations, it might be possible to use the L2D_CACHE event instead of L2D_CACHE_RD or L2D_CACHE_WR and the L2D_CACHE_REFILL event instead of L2D_CACHE_REFILL_RD or L2D_CACHE_REFILL_WR.

773437
External abort or uncorrected ECC error might cause further data corruption

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

An external abort or poisoned data on a linefill might cause data corruption on a later store instruction.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The core has entered write streaming mode because of a number of full cache line writes.
2. Either:
   - A Compare And Swap (CAS) instruction is executed.
   - Hardware pagetable updates of the Access Flag or Dirty Bit are enabled, and an instruction fetch or load or store instruction causes a hardware update.
3. The linefill started by the CAS or hardware pagetable update receives either an external abort or poisoned data from an uncorrectable ECC error.
4. A store instruction is executed to a different address in writeback cacheable memory.

Implications

The data from the store instruction might be lost, causing data corruption. Note that this requires an external abort or uncorrected ECC error, which already is likely to be fatal to the running process. Therefore the result is a small increase in the failure in time (FIT) rate.

Workaround

No workaround is required. For designs where RAS is of significant importance, this erratum can be avoided by disabling an optimization related to write streaming of store instructions. This will have a small impact on streaming performance. To do this, set CPUACTLR(_EL1)[41] to 1.

776820
ESR_ELx.WnR bit set incorrectly for some aborts on atomic instructions

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Aborts on v8.1 atomic instructions because the virtual address is out of range are reported with ESR_ELx.WnR bit set incorrectly.
The WnR bit should be set to 0 if a read of the address generates an abort, but this erratum causes the WnR bit to be incorrectly set to 1 for this type of abort.

Conditions

1. An atomic instruction is executed.
2. The MMU is on.
3. Bits [63:48] of the virtual address are not all zeros or all ones, causing a translation fault. If the relevant TCR.TBI bit is set, then bits [55:48] of the virtual address are not all zeros or all ones instead.

Implications

The WnR bit is not expected to be useful for this type of abort, therefore there are no implications for typical software.

Software workaround

No workaround is necessary.

777436
Speculative hardware page table updates might cause an abort when the stage 2 page tables are not cacheable

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The ARM architecture only requires hardware page table updates to be atomic when the page tables are held in certain memory types. For other memory types, the Cortex-A55 processor will generate an abort if a hardware update is required. On speculative transactions that turn out to not require a hardware update, the abort can be incorrectly reported.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. EL2 is AArch64.
2. VTCR_EL2.HA and VTCR_EL2.HD are both set.
3. VTCR_EL2.IRGN0 or VTCR_EL2.ORGN0 indicate a memory type that is not write-back memory.
4. The stage2 page table descriptor access permissions indicate read only, and the dirty bit modifier is set.

Implications

If the erratum occurs, a stage 2 abort will be reported on a translation that did not architecturally require a stage 2 hardware update.
Note that it is expected for performance reasons that stage 2 page tables would not normally be stored in non-cacheable memory. Additionally, hardware updates on non-speculative accesses would always abort when the memory type is not cacheable, therefore there is no benefit to setting the VTCR_EL2.HD bit in these cases.

Workaround

No workaround is believed to be necessary.

784215
ETM trace reports incorrect branch target

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

When executing a branch that targets an out of range virtual address, the ETM might trace the target address incorrectly.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. ETM trace is enabled and not prohibited.
2. The core is executing code located in the virtual address range 0hFFFF000000000000-0hFFFFFFFFFFFFFFFF, and bit[48] of the PC is set. If the relevant TCR.TBI bit is set, then bits[63:56] of the virtual address can be any value.
3. The core executes a branch instruction with the target virtual address in the range 0h0002000000000000-0hFFFEFFFFFFFFFFFF, with bit[48] of the target address clear. This could be any kind of instruction that alters the program flow, including immediate branches, register based branches, and returns.

Implications

The target of the branch will take a translation fault, because it is not in a valid address range, but because the ETM trace decompression will report the branch target incorrectly the reason for the fault might be harder to determine when debugging. Bits[47:0] of the branch target will still be correct, but bits[63:48] will be incorrectly reported as all zero.

Workaround

There is no workaround for this erratum.

789947
DVM might prevent a core power off

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a DVM message is received by a core at a specific time during a power off sequence, then it might cause the P-channel request to be incorrectly denied.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The CPUPWRCTLR.CORE_PWRDN_EN is set and a WFI instruction is executed.
2. A request is received on the core P-channel interface to put the core into the OFF or EMU_OFF power mode.
3. After the hardware clean of the data caches has completed, but before the power off is complete, the core receives a DVM snoop. This could be from another core in the cluster executing a TLB or instruction cache invalidate, or from the external interconnect.

Implications

If the above conditions occur, then the P-channel request would be incorrectly denied, and so the power off would not take place. However if the power controller then made a second request to put the core into the OFF or EMU_OFF power mode, it would complete successfully, provided that condition 3 did not also occur on this second attempt.
Note that this erratum is categorized assuming that the power controller is tolerant of getting a spurious deny, and will retry in these cases.

Workaround

The power controller should retry the P-channel request if it was unexpectedly denied.

795490
Incorrect fault status codes used for reporting synchronous uncorrectable ECC aborts

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

When reporting a synchronous external abort because of an uncorrectable ECC error, an incorrect encoding is written to the fault status code field of the status registers.

Configurations Affected

This erratum only affects configurations with ECC support enabled by setting CORE_CACHE_PROTECTION to TRUE or setting the DSU configuration SCU_CACHE_PROTECTION to TRUE.

Conditions

1. An uncorrectable ECC error occurs, either in the L1, L2, or L3 data cache RAMs, or (if the CHI bus interface is configured) in an external component that then returns poisoned data on the external bus interface.
2. A load instruction reads the poisoned data, and causes a synchronous external abort.

Implications

The status registers are updated with a fault status code that is a reserved value in the ARM v8.2 architecture. When the long descriptor page format is in use, the codes correspond to those that were used for synchronous parity or ECC errors in previous versions of the architecture. Error handling software will have to interpret the reserved values used.
When the short descriptor page format is in use, software cannot make a distinction between a Domain fault, level 1, or a synchronous external abort on memory access, as both will be reported as 0b01001.

Workaround

When reading the fault status code from the DFSC, IFSC, or FST fields of the ESR/HSR/FSR/PAR registers, software will have to be aware that the value reported might not be accurate.

- Case 1. The long descriptor format is used and software reads any of the following values:

- 0b011000
- 0b011100
- 0b011101
- 0b011110
- 0b011111

Software should clear bit [3] of the status code before interpreting the value.

- Case 2. The short descriptor format is used and software reads the value 0b01001.

Software could have been caused by a Domain fault, level 1, or a synchronous external abort. Software will have to determine if a domain fault could have occurred for this address.

801844
ECC error might cause silent data corruption

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Some store instructions need to do a read-modify-write sequence to update the ECC bits in the cache. If the read part of this sequence detects a single bit ECC error in the RAMs, then under certain timing conditions and store traffic patterns, the correction mechanism might cause the store to update the wrong cache line. This results in silent data corruption.

Configurations Affected

This erratum only affects configurations with ECC support enabled by setting CORE_CACHE_PROTECTION to TRUE.

Conditions

1. A byte or halfword store is executed.
2. One or more further stores are executed. These stores must be to the same cache line, but must be to a different 128-bit part of the cache line. When merged together, the stores must cover one or more aligned 32-bit quantities, but must not cover the full 128 bits.
3. A single bit ECC error is detected in the 128-bit part of the L1 data cache line that matches the address of the store in condition 1.
4. Another byte or halfword store is executed, to the same 128-bit part of the cache line as the store in condition 2.

Implications

There is still substantial benefit being gained from the ECC logic.
This erratum might cause a small increase in overall system failure rate.
The detection of the error is still reported in the error record registers if they are configured to count correctable errors.

Workaround

No workaround is required.

802429
Debug not entering Memory Access mode without setting EDSCR.ERR

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor supports entering debug Memory Access (MA) mode through the APB interface. This might fail when an instruction previously executed through the APB interface has not yet completed.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The debugger inserts an instruction through the APB interface by writing the EDITR.
2. The debugger writes to the EDSCR to enter MA mode.
3. The debugger reads the DTRTX at the same time as the instruction completes.

Implications

The APB read of the DTRTX does not start up the MA state machine, but EDSCR.TXU and EDSCR.ERR are not set.

Workaround

Before executing the instruction, the debugger should write 1 to the EDRCR.CSPA. After executing the instruction, the debugger should poll the EDSCR.PipeAdv bit to determine when the instruction has completed, and only attempt to enter MA mode after the PipeAdv bit is set.

804701
ESR_EL2.ISV bit set incorrectly for some external aborts

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The RAS extensions in the ARMv8.2 architecture require the ESR_ELx.ISV bit to be zero on all synchronous external aborts. This erratum will cause this bit to be set on synchronous external aborts caused by a stage 2 translation table walk.

Conditions

A stage 2 translation table walk causes a synchronous external abort.

Implications

It is not expected that any software will be relying on this bit being zero, therefore there are no implications.

Software workaround

No workaround is necessary.

804765
Mismatch between EDPRSR.SR and EDPRSR.R

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-A55 processor provides access to the EDPRSR through the APB interface. If this access is done at the same time as the core leaves a Warm reset, then a subsequent read of the same register will read an incorrect value of the SR field.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The core is in Warm reset.
2. A debugger reads the EDPRSR register over the APB interface.
3. The core comes out of Warm reset during the APB read.
4. A second APB read is made to the EDPRSR register.

Implications

The first read of the EDPRSR will read the SR field and R field both as 0b1.
The second read of the EDPRSR.SR field will read 0b0 whereas the previous read of the EDPRSR.SR was 0b1 while in Warm reset. Because the first read took place while in Warm reset, the sticky bit should still be set on the second read.

Workaround

If the debugger reads the EDPRSR and sees both the SR and R fields set, then it must remember this result and on the next EDPRSR read treat the SR bit as if it was set.

826172
Accessing the CLUSTERTHREADSIDOVR_EL1 register generates an UNDEFINED exception

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

The implementation-specific CLUSTERTHREADSIDOVR_EL1 register can be used by more privileged software, for example a hypervisor, to delegate some level of control of the L3 cache partitioning and bus QoS functionality to less privileged software. Because of this erratum, all accesses to the register take an UNDEFINED exception.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor. However, the L3 cache partitioning functionality is only present when used with the DSU revision r1p0 or higher.

Conditions

Software accesses the CLUSTERTHREADSIDOVR_EL1.

Implications

Access to the functionality provided by the CLUSTERTHREADSIDOVR_EL1 register is not available.

Software workaround

Only a limited combination of Cortex-A55 and DSU revisions is impacted. Therefore, no workaround is expected to be needed.

827496
Data cache maintenance operations by set/way targeting a 4MB L3 cache might affect incorrect cache index

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r1p0.

Description

Data cache maintenance by set/way instructions (DC CISW, DC CSW, DC ISW, and their AArch32 equivalents) encodes a set, way, and cache level. The largest cache supported by the Cortex-A55 processor is a 4MB Level 3 cache, which requires 12 bits to encode the cache set. For data cache maintenance instructions by set/way targeting this largest cache size, bit [11] of the set will always be treated as 0.

Configurations affected

This erratum only affects configurations of the Cortex-A55 processor with a 4MB L3 cache.

Conditions

1. The CPU executes a set/way data cache maintenance instruction targeting the L3 cache for which the MSB of the target index is 1.

Implications

Software attempting to clean and/or invalidate particular indexes from the L3 cache will be unable to do so. The contents of the cache are not corrupted by this erratum, but the data might not be visible to an external non-coherent agent as expected after the maintenance.
Note that in general cache maintenance by set/way is not suitable for use in a coherent memory system unless no masters are accessing the cache during the maintenance, therefore typical software will not be making use of these operations.
This erratum does not affect the hardware cache cleaning that is performed as part of a power off sequence.

Workaround

There is no workaround.

872696
Multiple stores to the same address targeting different types of device memory might cause data corruption

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

When executing loads and stores with mismatched device memory attributes, recent store data might not be observable to a subsequent load even though that store and load targeted the same type of device memory.

Configurations Affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. A store-release instruction is executed to virtual address A.
2. Virtual address A is marked in the pagetables as Device nGnRnE, Device nGnRE, or Device nGRE.
3. A store instruction is executed to virtual address B.
4. Virtual address B is marked in the pagetables as Device-GRE memory.
5. An LDAPR instruction is executed to virtual address B.
6. The LDAPR instruction must overlap some bytes of the store instruction to address B.
7. Virtual addresses A and B both map to the same 64-byte region of physical memory.

Note that the two stores could happen in either order.

Implications

If the conditions are met, then the load to address B might observe older data from address B, rather than the data from the earlier store to address B.

Workaround

No workaround is expected to be needed for most software. For any software that uses mismatched device memory attributes, it should put a DMB between any uses of the different attributes.

900307
Stage2 dirty bit not updated when HCR_EL2.DC set

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r1p0.

Description

When stage2 pagetable hardware updates of the dirty bit are enabled, and the HCR_EL2.DC bit is set, then under some conditions a store instruction might update the memory without updating the stage2 access permission.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

1. The hypervisor at EL2 is in AArch64 state.
2. The HCR_EL2.DC bit is set.
3. The VTCR_EL2.HA and VTCR_EL2.HD bits are both set.
4. The processor is executing code in AArch32 state at Non-secure EL0 or EL1.
5. The TTBCR.EAE bit is zero.
6. The DACR register has domain 0 set to 0b00 (no access).
7. A store instruction is executed.
8. The stage2 pagetable corresponding to the address of the store has the DBM bit set and S2AP[1] clear.
9. The pagetable entry is cached in the TLB because of a previous load or speculative access.

Implications

A store can write to memory without having stage2 write permission, and without triggering a hardware update of the pagetable. This cause data loss if the hypervisor later relies on the dirty bit status in the pagetable.

Workaround

If the hypervisor runs 32-bit guests with the HCR_EL2.DC bit set, then it should not enable hardware dirty bit updates on stage2 pagetables, and should instead use software handling of the dirty status of the page.

948532
Incorrect PMCEID1 value

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, and r1p0. Fixed in r2p0.

Description

The PMCEID1/PMCEID1_EL0 register indicates which PMU events are implemented. It incorrectly indicates that REMOTE_ACCESS is implemented and also incorrectly indicates that REMOTE_ACCESS_RD is not implemented.

Configurations Affected

All configurations are affected.

Conditions

The PMCEID1/PMCEID1_EL0 register is read and the contents of bits [17] or [24] are used.

Implications

Software trying to discover available PMU events will not find the REMOTE_ACCESS_RD event, but the event itself is still usable. Software might try and use the REMOTE_ACCESS event which is not implemented.

Workaround

Software should ignore the value of the two incorrect bits.

999993
Multiple concurrent ECC errors might cause silent data corruption

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0 and r2p0. Open.

Description

If multiple separate ECC errors are detected in the L1 data cache at around the same time, then in rare cases a store might write to an incorrect way and cause silent data corruption.

Configurations Affected

This erratum only affects configurations with ECC support enabled by setting CORE_CACHE_PROTECTION to TRUE.

Conditions

1. An ECC error is detected in the L1 data cache tag or data RAMs by a store, a pagewalk, or a prefetch.
2. One or more store instructions are executed so that:
   - More than one 16-byte aligned region within a single cache line is written, which must be a different line to the one on which the ECC error in condition 1 was detected.
   - At least one of the regions is only partially written.
3. A single-bit ECC error is detected within a partially written 16-byte region written by condition 2.

Implications

There is still substantial benefit being gained from the ECC logic.
This erratum might cause a small increase in overall system failure rate.
The detection of the errors is still reported in the error record registers if they are configured to count correctable errors.

Workaround

No workaround is required.

1017312
DBGDSCRext.SPNIDdis and DBGDSCRint.SPNIDdis might be erroneous

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, and r1p0. Fixed in r2p0.

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-A55 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.

1030596
Cycle count value in timestamp packet might be incorrect

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault: Status: Present in r0p0, r0p1, r1p0, and r2p0. 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.

1030597
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, r0p1, r1p0 and r2p0. 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.

1131793
Stores might prevent progress of an exclusive loop

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

If stores are included in a loop with a load and store exclusive sequence, then they might prevent the exclusive sequence from passing.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A loop including a load and store exclusive instruction is executed.
2. The loop contains additional store instructions before the load exclusive, or after the store exclusive, which get executed every time around the loop, including when the loop is re-executed because the store exclusive is failing.
3. The additional store instructions are at least two stores to the same cache line, and both stores are Cacheable. The stores can be to a different cache line from the load and store exclusive.
4. The stores must either:
    - Not allocate into the cache. This means that they are to memory marked as either Cacheable No Write-Allocate or Transient, or that the stores are non-temporal.
    - Be to the same cache line as an address that is being repeatedly read by another core.

Implications

The store exclusive sequence might continually fail, causing a software livelock. Interrupts can still be taken so the OS or hypervisor can break out of the loop.

Workaround

Non-allocating stores are not expected in typical exclusive loops. Such loops should be rewritten to use allocating stores. If the address being stored to is likely to be polled by another core, then the stores should be moved outside of the exclusive loop.

1214720
VFP_SPEC PMU event might count incorrectly

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

The Cortex-A55 core implements several architectural and IMPLEMENTATION DEFINED performance monitor events. Because of this erratum, the VFP_SPEC event might not count correctly.

Configurations Affected

All configurations with NEON_FP enabled are affected.

Conditions

One of the PMU event counters is configured to count the 0x075, VFP_SPEC event.

Implications

The PMU counts SIMD floating-point instructions as well as scalar floating-point instructions.

Workaround

In some situations, where integer SIMD instructions are known not to be used, the ASE_SPEC count can be subtracted from the VFP_SPEC count to give the number of scalar VFP instructions.

1401736
Non-cacheable loads of mismatched size might not be single-copy atomic

Status

Affects: Cortex-A55
Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r1p0 and r2p0. Open.

Description

The Cortex-A55 core supports single-copy atomic load and store accesses as described in the Arm architecture. However, in some unusual code sequences, this erratum can cause the core executing a store and later a load to the same address but with a different access size to load data that does not meet the requirements of a single-copy atomic load.

Configurations affected

All configurations are affected.

Conditions

On one core, the following condition must occur:

A store instruction is executed. This could be any halfword, word, or doubleword store instruction that is not a store release, and the address must be aligned to the access size. The store must be to Normal Non-cacheable, Normal Write-Through, or Device-GRE memory.

On a second core, which can be within the same cluster or in a different cluster, the following conditions must occur:

1. A store instruction is executed. This store must be a smaller access size to the store on the first core, and must be to an address within the bytes accessed by the first core. The address must also be aligned to the access size.
2. A load instruction is executed. The load must be a larger access size than the store from the same core, and at least some bytes of the load must be to the same address as the store. The address must also be aligned to the access size.

The Arm architecture requires that the load is single-copy atomic. However, in the conditions described, the load might observe a combination of the two stores, indicating that the store on the first core was serialized first. However, if the load is repeated, then the second time it might see just the data from the store from the first core indicating that the store on the first core was serialized second.

Implications

Concurrent, unordered stores are not common in multi-threaded code. In the C11 standard, they are restricted to the family of "relaxed" atomics. In addition, using different size load and store instructions to access the same data is unusual. Therefore the majority of multi-threaded software is not going to meet the conditions for this erratum.

Workaround

Most multi-threaded software is not expected to meet the conditions for this erratum and therefore does not require a workaround. If a workaround is required, then the store on the second core should be replaced with a store release instruction.

2109978
Atomic store instructions might not report an External abort or System Error

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

If the core receives read data from the interconnect where some beats of the data indicate an error, then, for certain types of atomic instruction, the core might not report the error.

Configurations Affected

This erratum affects all configurations.
However, if the system interconnect supports poisoning, then it is unlikely to meet the other conditions required.

Conditions

1. The CPUECTLR_EL1.ATOM field is set to 0b01, which forces all store atomics to be performed in the L1 cache.
2. A core executes an atomic store instruction to Inner Write-Back, Outer Write-Back cacheable memory.
3. The address accessed by the instruction is not present in any cache in the cluster, so the DSU sends a read transaction to the system interconnect.
4. The interconnect returns data that contains an error.
    - On CHI, this means some data flits indicate DERR or NDERR.
    - On ACE, this means some data transfers indicate SLVERR or DECERR.

Implications

If these conditions are met, then the core will correctly discard the data from the atomic store and the data from the interconnect, however it will not report an External abort or System Error interrupt. Therefore, the software will be unaware that the data was discarded.
The software is unlikely to use an atomic store as the first instruction to access this address. Therefore, if the error response from the interconnect is because of a permanent fault such as the address not being mapped to any peripheral, then an abort would have been reported when the software initially wrote to that address with a normal store instruction.
Systems using a CHI interface and configured with ECC support would be expected to poison data that got an uncorrectable error, rather than returning a DERR or NDERR. These systems using poison would not be impacted by this erratum.
In systems that meet the configurations described, this erratum might cause a negligible increase in overall system failure rate.

Workaround

For many systems that have not changed the default value of CPUECTLR_EL1.ATOM, no workaround is necessary. The software should not set the CPUECTLR_EL1.ATOM field to 0b01.

2288055
DTR flags not cleared on external debugger access while leaving Debug state

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

The Data Transfer Registers (DTRs) provide a mechanism to transfer data between an external debugger and the core. They consist of write-only registers to transmit data (DBGDTRTX_EL0 and DBGDTRTXint), read-only registers to receive data (DBGDTRRX_EL0 and DBGDTRRXint), and associated data control flags.

Due to this erratum, if these registers are accessed by the external debugger while the debug exit procedure is in progress, then the accesses will go ahead but the flow control flags will not be updated correctly.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The debugger requests a debug exit.
2. The debugger does an external access to the DBGDTRRX/ DBGDTRTX, while the debug exit is ongoing.
3. Certain microarchitectural timing conditions are met.

Implications

For an external read to DBGDTRTX, the EDSCR.TXU and EDSCR.TXfull flags will not be updated.
For an external write to DBGDTRRX, the write will be ignored and the EDSCR.RXO and EDSCR.RXfull flags will not be updated.

Workaround

There is no workaround.

2342475
Halting step syndrome might be wrong on stepping a Load-Exclusive instruction

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

Halting Step is a debug resource that a debugger can use to make the core step through code one instruction at a time. The EDSCR.STATUS records different scenarios for entering Debug state on a Halting Step debug event.

Because of this erratum, the EDSCR.STATUS might be wrong after entering Debug state on a Halting Step debug event.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

This erratum occurs when the following sequence of conditions is met:

1. The debugger activates Halting Step with the core in the Debug state
2. The debugger signals the core to exit Debug state
3. The core steps a Load-Exclusive instruction which generates an exception with the relevant SCTLR_ELx.IESB == 1
4. The core enters Debug state
5. The debugger reads EDSCR.STATUS

Implications

If this erratum occurs, the debugger might read EDSCR.STATUS value as 'Halting Step, normal' instead of the correct value as 'Halting Step, exclusive' or 'Halting Step, no syndrome'.

Workaround

There is no workaround.

2364958
Some unallocated debug and trace System registers might be trapped by HCR_EL2.TIDCP

Status

Affects: Cortex-A55
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

Description

In Cortex-A55, HCR_EL2.TIDCP is used to trap access to **`IMPLEMENTATION DEFINED`** System instructions or System registers from EL0 or EL1.

Because of this erratum, the HCR_EL2.TIDCP might trap some unallocated System registers in the debug and trace group with op0==0b10.

Configurations affected

This erratum affects all configurations of the Cortex-A55 processor.

Conditions

This erratum occurs when all the following conditions are met:

- HCR_EL2.TIDCP is set
- Software executes an MRS or MSR (register) with op0==0b10 and CRn=={11, 15} in EL1

Implications

If this erratum occurs, the access will be trapped by the HCR_EL2.TIDCP rather than raising an **`UNDEFINED`** exception.

Workaround

There is no workaround.