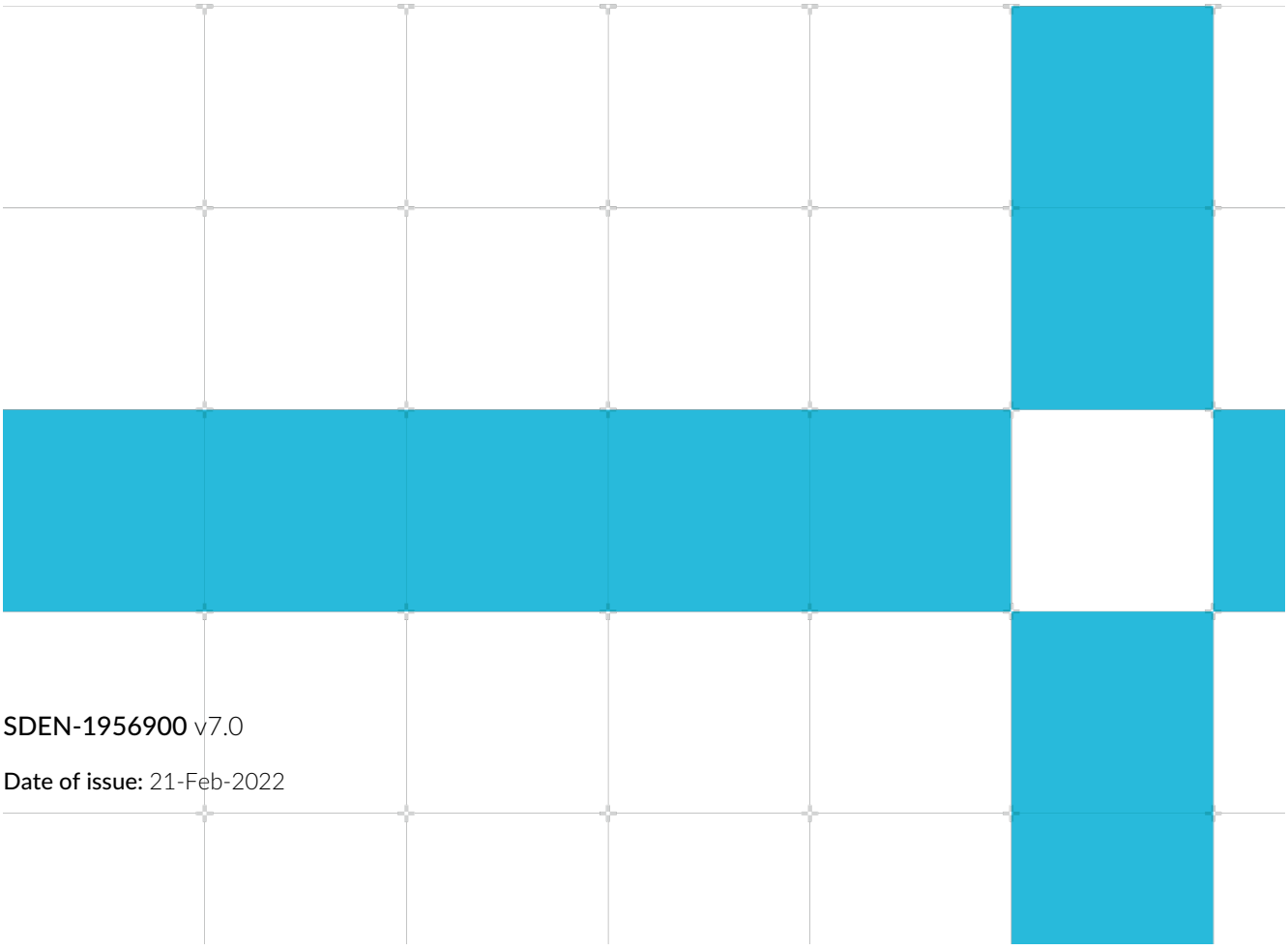
Arm Cortex-R82 Processor MP130

Software Developer Errata Notice

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



SDEN-1956900 v7.0

Date of issue: 21-Feb-2022

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 Arm Cortex-R82 Processor MP130 , 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

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.

21-Feb-2022: Changes in document version v7.0

ID	Status	Area	Category	Summary
2413308	New	Programmer	Category B	Read of CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2 might return the value from last interrupted write
2416441	New	Programmer	Category B	Clock is not gated correctly when the core is in the Full retention power mode
2416442	New	Programmer	Category C	Multiple double bit ECC errors on core powerdown might deadlock

13-Dec-2021: Changes in document version v6.0

ID	Status	Area	Category	Summary
2275980	New	Programmer	Category B	Cluster transitions to a lower power mode may cause deadlocks in the presence of Core transitions to higher power modes
2284596	New	Programmer	Category B	Interrupt can be routed to a Core in FULL_RET although SCLK Q-channel is not in QRUN
2295080	New	Programmer	Category B	Shared register writes do not respect context synchronization barriers
2313135	New	Programmer	Category C	Cacheable accesses to LLRAM might deadlock in the presence of an ECC error
2358852	New	Programmer	Category C	Pointer authentication key registers are not initialized
2359684	New	Programmer	Category C	Some unallocated debug and trace System registers might be trapped by HCR_EL2.TIDCP
2370219	New	Programmer	Category C	Use of debug related power states can result in deadlock
2375011	New	Programmer	Category C	PSTATE.SS not restored correctly if OSLK is set to 1 just before debug exit
2384696	New	Programmer	Category C	DPU returns wrong value on MRS to DLR_EL0
2386524	New	Programmer	Category C	Defective RAS registers

12-Nov-2021: Changes in document version v5.0

ID	Status	Area	Category	Summary
2309270	New	Programmer	Category B	Transition to FULL_RET is denied if IMP_CPUPWRCTLR_EL1.FPURET > 0
2324971	New	Programmer	Category B	Critical error interrupt will not be indicated for UC error on Error Records 1, 2 and 3
2329077	New	Programmer	Category B	Cluster PMU register interface has limited functionality
2330574	New	Programmer	Category B	Transition from L2 cache FULL RAM ON to DL1ONLY operating mode might cause a deadlock
2342667	New	Programmer	Category B	An ECC error during a power-down sequence might affect execution of FP/AdvSIMD instructions
2344782	New	Programmer	Category B	DSB might not ensure completion of previous stores when multiple contexts are using the same port
2284986	Updated	Programmer	Category C	PMU event counts might be inaccurate
2317561	New	Programmer	Category C	Halting step syndrome might be wrong on stepping a Load-Exclusive instruction
2330237	New	Programmer	Category C	Multi-issuing control with IMP_CPUACTLR_EL1.MI = 0b010 is wrong
2338818	New	Programmer	Category C	IFU Cache errors might report incorrect BANK to the RAS node
2339884	New	Programmer	Category C	Cluster CTI trigger outputs are not reaching cluster ELA CTITRIGINs

23-Sep-2021: Changes in document version v4.0

ID	Status	Area	Category	Summary
2290922	New	Programmer	Category B	Link register might be incorrect after taking an SError interrupt exception or entering Debug state
2284986	New	Programmer	Category C	PMU event counts might be inaccurate
2286314	New	Programmer	Category C	Access to TRCSEQEVR3 is not UNDEFINED

19-Aug-2021: Changes in document version v3.0

ID	Status	Area	Category	Summary
2272083	New	Programmer	Category B	Utility bus accesses to RAS registers target wrong registers
2151159	New	Programmer	Category C	Multi-Core Cortex-R82 may deadlock when executing L2DBG operations and WFx events
2173953	Updated	Programmer	Category C	PMU event counts might be inaccurate
2224832	New	Programmer	Category C	Debug APB accesses to reserved core address is accessible when core powered off
2246499	New	Programmer	Category C	Some RAS and Debug identification registers have wrong values
2248150	New	Programmer	Category C	Accesses to non-secure EL2 physical timer registers and secure EL2 virtual timer registers are not UNDEFINED
2254257	New	Programmer	Category C	Read of ERRGSR might return incorrect value
2261658	New	Programmer	Category C	A single bit hard error on poisoned TCM data could cause a deadlock

27-Jul-2021: Changes in document version v2.0

ID	Status	Area	Category	Summary
2215640	New	Programmer	Category A	A cacheable write to LLRAM could cause data corruption of a Main Master cacheline
2113395	New	Programmer	Category B	Incorrect value could be reported in AFSR0_ELx for external errors on store-exclusive and load atomic instructions
2114617	New	Programmer	Category B	Cortex-R82 could deadlock in the presence of an L2 Cache Data RAM ECC Error
2130548	New	Programmer	Category B	An ECC error could cause a deadlock under certain conditions
2177521	New	Programmer	Category B	Physical interrupts might not be taken
2188383	New	Programmer	Category B	ECC error might cause wrong execution of certain system register access instructions
2110865	New	Programmer	Category C	RAM ECC errors or poison during core power off transitions may lead to deadlock
2131519	New	Programmer	Category C	RAS Fault Injection to shared memory may cause deadlock or data corruption
2131811	New	Programmer	Category C	DSB does not guarantee the observability of the effects of the GIC CPU interface register accesses
2140942	New	Programmer	Category C	Incorrect value reported in RAS registers on L2 cache ECC errors
2169550	New	Programmer	Category C	ETM trace might not report certain direct branch instructions
2170009	New	Programmer	Category C	Error injection via ERR0PFGCTL might occur at the wrong time
2173953	New	Programmer	Category C	PMU event counts might be inaccurate
2215960	New	Programmer	Category C	An atomic with acquire semantics may be observed before an earlier store-release to TCM or LLPP
2218751	New	Programmer	Category C	Data cache maintenance operations by set/way targeting a 4MB L2 cache might affect incorrect cache index
2226045	New	Programmer	Category C	ATB flush is unreliable

26-Mar-2021: Changes in document version v1.0

No errata in this document version.

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
2215640	Programmer	Category A	A cacheable write to LLRAM could cause data corruption of a Main Master cacheline	r0p0	r0p1
2113395	Programmer	Category B	Incorrect value could be reported in AFSR0_ELx for external errors on store-exclusive and load atomic instructions	r0p0	r0p1
2114617	Programmer	Category B	Cortex-R82 could deadlock in the presence of an L2 Cache Data RAM ECC Error	r0p0	r0p1
2130548	Programmer	Category B	An ECC error could cause a deadlock under certain conditions	r0p0	r0p1
2177521	Programmer	Category B	Physical interrupts might not be taken	r0p0	r0p1
2188383	Programmer	Category B	ECC error might cause wrong execution of certain system register access instructions	r0p0	r0p1
2272083	Programmer	Category B	Utility bus accesses to RAS registers target wrong registers	r0p0	r0p1
2275980	Programmer	Category B	Cluster transitions to a lower power mode may cause deadlocks in the presence of Core transitions to higher power modes	r0p0, r0p1	r0p2
2284596	Programmer	Category B	Interrupt can be routed to a Core in FULL_RET although SCLK Q-channel is not in QRUN	r0p0, r0p1	r0p2
2290922	Programmer	Category B	Link register might be incorrect after taking an SError interrupt exception or entering Debug state	r0p0, r0p1	r0p2
2295080	Programmer	Category B	Shared register writes do not respect context synchronization barriers	r0p0, r0p1	r0p2
2309270	Programmer	Category B	Transition to FULL_RET is denied if IMP_CPUPWRCTLR_EL1.FPURET > 0	r0p0, r0p1	r0p2
2324971	Programmer	Category B	Critical error interrupt will not be indicated for UC error on Error Records 1, 2 and 3	r0p0, r0p1	r0p2
2329077	Programmer	Category B	Cluster PMU register interface has limited functionality	r0p0, r0p1	r0p2
2330574	Programmer	Category B	Transition from L2 cache FULL RAM ON to DL1ONLY operating mode might cause a deadlock	r0p0, r0p1	r0p2
2342667	Programmer	Category B	An ECC error during a power-down sequence might affect execution of FP/AdvSIMD instructions	r0p0, r0p1	r0p2
2344782	Programmer	Category B	DSB might not ensure completion of previous stores when multiple contexts are using the same port	r0p0, r0p1	r0p2
2413308	Programmer	Category B	Read of CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2 might return the value from last interrupted write	r0p0, r0p1, r0p2	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
2416441	Programmer	Category B	Clock is not gated correctly when the core is in the Full retention power mode	r0p2	Open
2110865	Programmer	Category C	RAM ECC errors or poison during core power off transitions may lead to deadlock	r0p0	r0p1
2131519	Programmer	Category C	RAS Fault Injection to shared memory may cause deadlock or data corruption	r0p0	r0p1
2131811	Programmer	Category C	DSB does not guarantee the observability of the effects of the GIC CPU interface register accesses	r0p0	r0p1
2140942	Programmer	Category C	Incorrect value reported in RAS registers on L2 cache ECC errors	r0p0	r0p1
2151159	Programmer	Category C	Multi-Core Cortex-R82 may deadlock when executing L2DBG operations and WFx events	r0p0	r0p1
2169550	Programmer	Category C	ETM trace might not report certain direct branch instructions	r0p0	r0p1
2170009	Programmer	Category C	Error injection via ERR0PFGCTL might occur at the wrong time	r0p0	r0p1
2173953	Programmer	Category C	PMU event counts might be inaccurate	r0p0	r0p1
2215960	Programmer	Category C	An atomic with acquire semantics may be observed before an earlier store-release to TCM or LLPP	r0p0	r0p1
2218751	Programmer	Category C	Data cache maintenance operations by set/way targeting a 4MB L2 cache might affect incorrect cache index	r0p0	r0p1
2224832	Programmer	Category C	Debug APB accesses to reserved core address is accessible when core powered off	r0p0	r0p1
2226045	Programmer	Category C	ATB flush is unreliable	r0p0	r0p1
2246499	Programmer	Category C	Some RAS and Debug identification registers have wrong values	r0p0	r0p1
2248150	Programmer	Category C	Accesses to non-secure EL2 physical timer registers and secure EL2 virtual timer registers are not UNDEFINED	r0p0	r0p1
2254257	Programmer	Category C	Read of ERRGSR might return incorrect value	r0p0	r0p1
2261658	Programmer	Category C	A single bit hard error on poisoned TCM data could cause a deadlock	r0p0	r0p1
2284986	Programmer	Category C	PMU event counts might be inaccurate	r0p0, r0p1	r0p2
2286314	Programmer	Category C	Access to TRCSEQEVR3 is not UNDEFINED	r0p0, r0p1	r0p2
2313135	Programmer	Category C	Cacheable accesses to LLRAM might deadlock in the presence of an ECC error	r0p0, r0p1	r0p2
2317561	Programmer	Category C	Halting step syndrome might be wrong on stepping a Load-Exclusive instruction	r0p0, r0p1	r0p2
2330237	Programmer	Category C	Multi-issuing control with IMP_CPUACTLR_EL1.MI = 0b010 is wrong	r0p0, r0p1	r0p2

ID	Area	Category	Summary	Found in versions	Fixed in version
2338818	Programmer	Category C	IFU Cache errors might report incorrect BANK to the RAS node	r0p0, r0p1	r0p2
2339884	Programmer	Category C	Cluster CTI trigger outputs are not reaching cluster ELA CTITRIGINs	r0p0, r0p1	r0p2
2358852	Programmer	Category C	Pointer authentication key registers are not initialized	r0p0, r0p1	r0p2
2359684	Programmer	Category C	Some unallocated debug and trace System registers might be trapped by HCR_EL2.TIDCP	r0p0, r0p1	r0p2
2370219	Programmer	Category C	Use of debug related power states can result in deadlock	r0p0, r0p1	r0p2
2375011	Programmer	Category C	PSTATE.SS not restored correctly if OSLK is set to 1 just before debug exit	r0p0, r0p1	r0p2
2384696	Programmer	Category C	DPU returns wrong value on MRS to DLR_EL0	r0p0, r0p1	r0p2
2386524	Programmer	Category C	Defective RAS registers	r0p0, r0p1	r0p2
2416442	Programmer	Category C	Multiple double bit ECC errors on core powerdown might deadlock	r0p2	Open

Errata descriptions

Category A

2215640
A cacheable write to LLRAM could cause data corruption of a Main Master cacheline

Status

Affects: Cortex-R82
Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor has a *Main Master* (MM) port and a *Low-latency RAM* (LLRAM) port that can both be cached in the L1 data cache. When both ports are used by software without intervening barriers, this erratum could cause corruption of data held in the L1 data cache for the MM port.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 processor with **NUM_CORES** >= 2 and the LLRAM port configured (**CFGLLRAMIMP**=1) and in use.

Conditions

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

1. A store to the LLRAM port is executed and the cacheline for this data is stored in the L1 data cache
2. Another core in the cluster performs a read of the cacheline for the LLRAM store, setting the cacheline to a shared state
3. A linefill to the MM port occurs to the same set and way as the LLRAM cacheline
4. When in a shared state, the LLRAM cacheline is invalidated from the L1 data cache. This could occur due to:
    - An *Error Correcting Code* (ECC) error in the *LLRAM Coherency Unit* (LCU) duplicate L1 tag RAMs
    - The linefill to the MM port to the same set and way above
5. Another store is executed to the same LLRAM cacheline at the same time as the cacheline is invalidated, but to a different 128-bit aligned region from the original store
6. The above conditions occur along with specific micro-architectural conditions with precise timing

Implications

When this erratum occurs, the contents of a cacheline in the L1 data cache holding data for the MM port could be overwritten with incorrect data.

Workaround

There is no workaround for this erratum.

Category A (rare)

There are no errata in this category.

Category B

2113395
Incorrect value could be reported in AFSR0_ELx for external errors on store-exclusive and load atomic instructions

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor reports additional information on faults in the AFSR0_EL1 and AFSR0_EL2 registers. This includes information about the memory port that the fault is associated with and if the fault is associated with data or not.

If an external uncorrectable *Error Correcting Code* (ECC) error occurs on the *Main Master* (MM) port, an atomic instruction to Normal cacheable memory targeting the MM port should report the fault type as Data (AFSR0_ELx.TYPE = 0b0000). However when this erratum occurs, the fault type is always reported as other (AFSR0_ELx.TYPE = 0b0011).

If an external error occurs on the MM port, an exclusive instruction to Normal cacheable memory targeting the MM port should report the port as MM (AFSR0_ELx.PORT = 0b0000). However when this erratum occurs, the port is always reported as **UNKNOWN** (AFSR0_ELx.PORT = 0b0111).

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when all of the following conditions are met:

- A core executes a store-exclusive instruction or a load atomic instruction (for example where the destination register is not the zero register)
- The store-exclusive instruction or the load atomic instruction is to the MM port with Normal Inner Write-Back, Outer Write-Back Cacheable attributes
- The data is not contained in the L1 data cache
- If the instruction is a load atomic instruction and all of the following are true:
  - The data is contained in the L2 cache
  - The data has an uncorrectable ECC error
- If the instruction is a store-exclusive instruction and one of the following conditions is true:

- The data is contained in the L2 cache and has an uncorrectable ECC error
- The data is not in the L2 cache and receives an external error in response to the bus request on the MM port

Implications

If this erratum occurs, then the value reported in the AFSR0_ELx register could be incorrect and be more general than intended.

Workaround

To avoid this erratum, software can confirm:

- If the port is affected by this erratum by using the information in the FAR_ELx register. The information in the FAR_ELx register can be used to show the address the error occurred on and compare it against the system memory map.
- If a data error has occurred by reading the RAS record error registers 3-5

2114617
Cortex-R82 could deadlock in the presence of an L2 cache data RAM ECC error

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor supports RAM protection for single bit and double bit *Error Correcting Code* (ECC) errors on the L2 cache data RAMs. In some cases, an ECC error in the L2 cache data RAMs causes the processor to deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- RAM protection is enabled (IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN=1)
- The L2 cache data RAM internal data buffering is full
- Both read and write accesses to the L2 cache data RAM are ongoing
- A single bit or double bit ECC error occurs on a L2 cache data RAM read request

Implications

If this erratum occurs, the processor deadlocks.

Workaround

To avoid this erratum, disable the RAM protection for shared memories by setting the IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN to 0.

Because this erratum occurs when there is an ECC error and because the ECC errors are not expected in sample silicon, it is still practical to evaluate the performance on sample silicon.

2130548
An ECC error could cause a deadlock under certain conditions

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor has dedicated hardware to forward data from a load instruction to a subsequent load or store address to decrease load-to-use latency. The processor also supports a load and store instruction to be issued together.

Under certain conditions, the combination of these features and an *Error Correcting Code* (ECC) error being detected in the L1 data cache, *Instruction Tightly Coupled Memory* (ITCM), or *Data Tightly Coupled Memory* (DTCM) can cause the processor to deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- RAM protection is enabled (IMP_MEMPROTCTLR_EL1.MEMPROTEN=1)
- The following instructions exist in a program:
  - A load instruction A
  - A load instruction B
  - A store instruction C
- The destination register of the load instruction A is the same as the address register for the store instruction C. Intermediate instructions between A and C do not modify this register.
- Instructions A and B are consecutive in program order, or only have instructions that are not load/store instructions between them. If the instructions A and B are not consecutive, there must be less than eight instructions between them.
- Instructions B and C are consecutive in program order, or have a single instruction between them which can be triple-issued with instructions B and C.
- There is an ECC error in the 128-bit aligned region of memory targeted by the load instruction A, and one of the following applies:
  - The ECC error is suppressed by data forwarded by a store instruction preceding write to the same address
  - The ECC error occurs in a word that is not accessed by the load instruction

- The load instruction B does not take a synchronous abort

Implications

If this erratum occurs, the processor deadlocks.

Workaround

To avoid this erratum, set the IMP_CPUACTLR_EL1.FPC to 0. This has a slight performance impact for some workloads.

Because this erratum occurs when there is an ECC error and because the ECC errors are not expected in sample silicon, it is still practical to evaluate the performance with IMP_CPUACTLR_EL1.FPC=1 for sample silicon.

2177521
Physical interrupts might not be taken

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements the *Generic Interrupt Controller* (GIC) CPU interface.

Because of this erratum, under certain conditions which involve heavy interrupt load, a processor core may stop taking new physical interrupts.

Configurations affected

The erratum affects all configurations of the Cortex-R82 processor when used with the GIC-625 interrupt controller.

Conditions

This erratum occurs when all the following conditions are met:

- The GIC CPU interface is enabled (**GICCDISABLE** signal LOW at reset)
- Group 1 physical interrupts are enabled
- The core has taken no less than four Group 1 physical interrupts, and before each of these there was a close-by physical interrupt acknowledged by the core

Implications

If this erratum occurs, the core might stop taking new physical interrupts.

Workaround

To avoid the erratum, set GICR0_FCTLR.ECP to 0 to disable combined packets in the Redistributor before waking-up the cores. This might increase the interrupt latency in certain cases.

2188383
ECC error might cause wrong execution of certain System register access instructions

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements the *Reliability*, *Availability*, *and Serviceability* (RAS) Extension as defined in the Arm architecture.

Because of this erratum, the read of certain System registers might return the wrong value or the write might be ignored after a change to the error status registers for the error records 0 to 2.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- Core RAM protection is enabled with IMP_MEMPROTCTLR_EL1.MEMPROTEN = 1
- RAS error reporting is enabled with ERR0CTLR.ED = 1
- One of the ERR<n>STATUS.V, where n is 0 to 2, is changed due to either a new *Error Correcting Code* (ECC) error or a software write
- At the same time, the core also executes a read/write to a System register belonging to one of the trace registers, the generic timer registers, the IMP_CLUSTER* registers, or the RAS Error Record registers 3 to 5

Implications

If this erratum occurs, the core might read **UNKNOWN** value from the System register or the write might be ignored.

Workaround

To avoid this erratum for the test silicon, set the ERR0CTLR.ED to 0 to disable RAS error reporting.

2272083
Utility bus accesses to RAS registers target wrong registers

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements the *Reliability*, *Availability*, *and Serviceability* (RAS) Extension as defined in the Arm architecture. The Cortex-R82 processor implements 6 error records.

Because of this erratum, the accesses to the RAS registers in error records 0 to 2 over the Utility bus will target wrong registers.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when software accesses to the RAS registers in error records 0 to 2 over the Utility bus.

Implications

If this erratum occurs, software might read incorrect values or will write to wrong RAS registers in error records 0 to 2.

Workaround

To avoid this erratum, software can shift the address offset right by 2 when accessing RAS registers in error records 0 to 2.

2275980
Cluster transitions to a lower power mode may cause deadlocks in the presence of Core transitions to higher power modes

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements individual power domains at the cluster and core levels. Because of this erratum, cluster transitions to a lower power mode may deadlock in the presence of simultaneous Core transitions to a higher power mode.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- The Cluster is transitioning to a lower power mode, particularly, ON->OFF or ON->MEM_RET.
- A Core in the system is requested to transition to a higher power mode, particularly, OFF->ON.
- The request overlap, and additional complex microarchitectural timing conditions occur.

Implications

When this erratum occurs, the system deadlocks.

Workaround

- For systems without a System Control Processor (or an equivalent agent capable of programming the PPU):
  - Set IMP_CPUPWRCTLR_EL1.PWRDN == 0 for at least 1 core in the system.

- For systems with a System Control Processor (or an equivalent agent capable of programming the PPU):
  - For static power transitions:
    - Ensure a previous transition has completed by polling the associated PPU_PWSR.PWR_STATUS (or CLUSTERPPU_PWSR.PWR_STATUS) and ensuring the

previously requested power transition completed before issuing a power mode transition to another device.

- For dynamic power transitions, it is impossible to use both Core and Cluster simultaneously, but they can be used individually:
  - To use dynamic power modes in a core, disable cluster dynamic power transitions CLUSTERPPU_PWPR.PWR_DYN_EN == 0.
  - To use dynamic power modes in the cluster, when the cores are OFF:
    - Disable core dynamic power transitions by setting PPU_PWPR.PWR_DYN_EN == 0 for all cores, enable cluster dynamic power transitions CLUSTERPPU_PWPR.PWR_DYN_EN == 1.
    - To restore a core to ON (e.g., to answer interrupts), the cluster dynamic power transitions must first be disabled (CLUSTERPPU_PWPR.PWR_DYN_EN == 0).

2284596
Interrupt can be routed to a Core in FULL_RET although SCLK Q-channel is not in QRUN

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements an SCLK Q-channel that allows SCLK to be gated when all cores are in either OFF or FULL_RET power modes. The processor can also receive external interrupts while in this power mode.

Because of this erratum, in the presence of an interrupt to a core in FULL_RET, the system may deadlock.

Configurations affected

This erratum affects configurations with Full Retention (auxiliary configuration parameter FULL_RET set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- All cores are in FULL_RET.
- SCLK Q-channel is in QSTOPPED mode - but SCLK is still present.
  - If the SCLK Q channel controller used in the design is one that does not provide SCLK at any time during QSTOPPED, this erratum does not apply.
- An interrupt arrives in the system.

Implications

When this erratum occurs, the system deadlocks.

Workaround

To avoid this erratum, disable the SCLK Q channel by setting IMP_CLUSTERACTLR_EL1.SCLKQ to 0.

2290922
Link register might be incorrect after taking an SError interrupt exception or entering Debug state

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

According to the Arm architecture, the link register (ELR_ELx or DLR_EL0) value should reflect the address to return to after taking an SError interrupt exception or restart address after entering the Debug state. The link register value should also account for any already executed instructions.

Because of this erratum, the link register value might not be set to the preferred return address after taking an SError interrupt exception or the preferred restart address after entering the Debug state.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when all the following conditions are met:

- A load instruction and a store instruction are dual-issued
- The older access is a store or a load that has completed
- One of the following occurs before the younger access has completed:
  - Debug state is entered
  - An SError interrupt exception is taken

Implications

If this erratum occurs, the value of the link register will indicate the older instruction that has already been executed. If returning from the SError interrupt exception or exiting the Debug state using the same value of the link register, the instruction pointed to by the link register will be executed including repeating any memory accesses by the instruction.

- If the memory access is to Device memory this could lead to a repeated access.
- If the memory access is a write to Normal memory this could violate single-copy atomicity.

Workaround

No workaround is expected to be required for the implications from taking an SError interrupt exception. This is because the Cortex-R82 processor will enter the Uncontainable error state after taking an SError interrupt exception so no recovery will be possible. If more information is required, then contact Arm support for more details.

To avoid the implications from entering Debug state, software can set IMP_CPUACTLR_EL1.LSDI to 1 to disable dual-issuing a load with a store instruction. This will have a small performance impact so should be used only when doing external debug.

2295080
Shared register writes do not respect context synchronization barriers

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements a set of system registers outside of the core power domain that software can access. A context synchronization event should create a barrier that ensures all system register writes are observable after the barrier.

Because of this erratum, for a set of system registers outside the core, this barrier is not be respected.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met:

- A core sends a write to one of the following system registers:
  - Any register whose name starts with IMP_CLUSTER*
  - Any RAS register relevant to the Shared Nodes 0, and 4-6
- A context synchronization event happens
- An instruction is executed that would observe the system configuration changes
- Additional complex microarchitectural conditions occur

Implications

When this erratum occurs, the effects of the system register write are not guaranteed to be observed after the context synchronization event.

Workaround

To work around this erratum, when there are changes to any IMP_CLUSTER* register or RAS register relevant to the Shared Nodes 0, and 4-6 and before executing instructions that are expected to observe the changes to the system registers software should read IMP_CLUSTERCFR_EL1.

2309270
Transition to FULL_RET is denied if IMP_CPUPWRCTLR_EL1.FPURET > 0

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, 0p1. Fixed in r0p2..

Description

In cases where both Full Retention (FULL_RET) and Functional Retention (FUNC_RET) are enabled in IMP_CPUPWRCTLR_EL1, it is not possible to transition to Full Retention (FULL_RET).

Configurations affected

This erratum affects configurations of the Cortex-R82 processor that have the Advanced SIMD and floating-point support enabled (NEON_FPm == 1) and support both Functional Retention and Full Retention (auxiliary configuration parameters FUNC_RET and FULL_RET both set to 1).

Conditions

This erratum occurs when the following conditions are met:

- IMP_CPUPWRCTLR_EL1.FPURET > 0
- IMP_CPUPWRCTLR_EL1.WFIRET > 0 or IMP_CPUPWRCTLR_EL1.WFERET > 0
- The core enters WFE or WFI
- The timer ticks configured in IMP_CPUPWRCTLR_EL1 are such where both FUNC_RET and FULL_RET are possible

Implications

If this erratum occurs, the core would be incapable of moving into FULL_RET.

Workaround

Whenever entering Full Retention (FULL_RET) is desirable, IMP_CPUPWRCTLR_EL1.FPURET must be set to 0 either permanently (disabling functional retention altogether) or temporarily right before executing WFI/WFE.

2324971
Critical error interrupt will not be indicated for UC error on Error Records 1, 2 and 3

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor supports the Critical error interrupt from the *Reliability, Availability, and Serviceability* (RAS) Extension. Due to this erratum a critical error interrupt will not be generated if an *Uncontainable Error* (UC) occurs in one of the components associated with Error Record 1, 2 and 3.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- Critical error interrupts are enabled for error records 1, 2 and 3 (**ERR1CTLR.CI** is set to 1).
- A double bit error occurs on the L1 data cache dirty RAM.

Implications

When this erratum occurs, no critical error interrupt will be generated.

Workaround

In order to work around this erratum, the critical error interrupt should not be enabled and set **ERR1CTLR.UI** to 1 to enable the error recovery interrupt. On handling an error recovery interrupt the error record register will indicate if a critical error has occurred in **ERR<n>STATUS.CI**.

2329077
Cluster PMU register interface has limited functionality

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

Cortex-R82 includes an **IMPLEMENTATION DEFINED** Cluster *Performance Monitoring Unit* (Cluster PMU) that monitors events in the processor cluster. The Cluster PMU can be accessed either by software running on any of the processor cores using System registers, or by an external agent through the Debug port.

Because of this erratum, the Cluster PMU functionality is limited.

Configurations affected

This erratum affects all configurations.

Conditions

In the case of Cluster PMU accesses through the Debug port, this erratum occurs when one of the following conditions is met:

- Reads of any register with a name starting with CLUSTERPMU_
- Writes to CLUSTERPMU_PMEVCNTR<n>_EL1, CLUSTERPMU_PMOVSSET_EL1, CLUSTERPMU_PMOVSCLR_EL1, CLUSTERPMU_PMINTENSET_EL1, or CLUSTERPMU_PMINTENCLR_EL1
- Resetting the counters by writing 1 to CLUSTERPMU_PMCR_EL1.P

In the case of Cluster PMU accesses through the **IMPLEMENTATION DEFINED** System registers, this erratum occurs when one of the following conditions is met:

- Reads of any register with a name starting with IMP_CLUSTERPM
- Resetting the counters by writing 1 to IMP_CLUSTERPMCR_EL1.P

Implications

When this erratum occurs:

- Reads of the registers that are in the Conditions of this erratum will be incorrect

- Writes to the registers that are in the Conditions of this erratum will have no effect

Workaround

For accesses through the Debug port:

- Reading any of the following six registers CLUSTERPMU_PMEVTYPER<n>EL1, CLUSTERPMU_PMOVSSET_EL1, CLUSTERPMU_PMOVSCLR_EL1, CLUSTERPMU_PMINTENSET_EL1, CLUSTERPMU_PMINTENCLR_EL1, CLUSTERPMU_PMCR_EL1, is not possible. To read any other register with a name that starts with CLUSTERPMU_, first write 0 to CLUSTERPMU_PMEVTYPER<n>EL1 (n= 0-5), and then perform the read.
- Resetting the counters through CLUSTERPMU_PMCR_EL1.P or other direct writes to the counters is not possible. A functionally equivalent behavior can be obtained by, reading the counter (bearing in mind the conditions of workaround 1) before event sampling is started and after it completed, and then performing a difference to obtain the counted value.
- Memory mapped writes to CLUSTERPMU_PMOVSSET_EL1, CLUSTERPMU_PMOVSCLR_EL1, CLUSTERPMU_PMINTENSET_EL1, CLUSTERPMU_PMINTENCLR_EL1 are not possible.
- The EDITR can be used to execute MSR and MRS instructions while the PE is in debug state which use the workarounds listed below to perform System register accesses to the affected registers.

For accesses through the System registers:

- To read any register with a name that starts with IMP_CLUSTERPM, first write 0 to IMP_CLUSTERPMXEVTYPER_EL1 for the selected IMP_CLUSTERPMSELR_EL1.SEL
- Instead of resetting the counters by writing 1 to IMP_CLUSTERPMCR_EL1.P, write 0 to IMP_CLUSTERPMXEVCNTR_EL1 while iterating all 6 counters on IMP_CLUSTERPMSELR_EL1.SEL

2330574
Transition from L2 cache FULL RAM ON to DL1ONLY operating mode might cause a deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor supports operating modes which allow powering down specific components. This includes the DL1ONLY operating mode where the L2 cache is powered down with supporting shared logic still active. If there is a stash request ongoing when the operating mode is requested to transition to DL1ONLY it could cause the processor to deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor with an L2 cache (**L2_CACHE_SIZE** > 0).

Conditions

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

1. A stash request occurs from a core targeting an address in the L2 cache.
2. A transition from FULL RAM ON to DL1ONLY is requested while the stash is being handled.

Implications

If this erratum occurs, the Cortex-R82 processor deadlocks.

Workaround

In order to workaround this erratum, the DL1ONLY operating mode should not be used. This will have a small impact on power as the L2 cache will remain powered up, but a transition to the OFF power mode is not affected by this erratum so it is still possible to power down the entire cluster.

2342667
An ECC error during a power-down sequence might affect execution of FP/AdvSIMD instructions

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements the *Reliability*, *Availability*, *and Serviceability* (RAS) Extension as defined in the Arm architecture. The processor has the capability of detecting *Error Correcting Code* (ECC) errors in RAMs and flagging an external RAS IRQ interrupt.

The processor can also power down its cores. If the processor generates an error-triggered RAS interrupt during a core power off sequence, it will abort and revert the power down to preserve the error information.

Because of this erratum, if a power-down transition is aborted due to a RAS interrupt the processor will deadlock when attempting to execute any floating-point or Advanced SIMD instruction at any point after the power transition is aborted.

Configurations affected

This erratum affects configurations with NEON_FP = 1 and RAM_PROTECTION = 1.

Conditions

This erratum occurs when all the following conditions are met:

- ERR<n>CTLR is configured to generate an interrupt.
- Error reporting is enabled (ERR<n>CTLR.ED = 1).
- A request to transition a core to the OFF power mode is underway.
- An ECC error (real or injected) occurs, which causes the power transition to be aborted.
- The processor attempts to execute a floating-point or Advanced SIMD instruction at any point after the power transition aborts.

Implications

If this erratum occurs, the processor deadlocks.

Workaround

To avoid this erratum, set the CI, DUI, CFI, FI, and UI fields in the ERR<n>CTLR to 0 before requesting a transition to the OFF power mode for any core.

2344782
DSB might not ensure completion of previous stores when multiple contexts are using the same port

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor provides multiple memory ports for different requirements. These ports have differing real-time characteristics and may be used by a single context or multiple contexts depending on the requirements of a particular system.

Due to this erratum, if the *Main Master* (MM) or *Low-latency RAM* (LLRAM) ports are shared across multiple contexts, then a DSB may complete before all previous stores that were performed to the shared port have completed.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

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

1. Store(s) are performed in a context to the MM or LLRAM ports
2. The context is changed, this could be:
    - A change between EL0/EL1 and EL2
    - Changing the value of VSCTLR_EL2.VMID
3. Store(s) to the same port are performed without an intervening barrier
4. A DSB that is affecting stores is executed before the stores from the old context have completed

Implications

When this erratum occurs, the DSB may complete before all previous outstanding stores have completed. It is expected that in the typical use case for Cortex-R82 the hypervisor code at EL2 will be in *Tightly Coupled Memories* (TCMs) to ensure a real-time response and so this erratum would not occur.

For a context switch between multiple guest operating systems, the expected latency of the context switch means that, in most cases, previous outstanding stores will be completed prior to starting execution of a different guest operating system and this erratum will not occur.

Workaround

To avoid this erratum, a DFB can be inserted when a context switch is performed to a guest operating system which uses the MM port or LLRAM port.

Note that a DFB should not be inserted when changing to a real-time context that does not use these ports to ensure that the latency of a context switch to a real-time context is not impacted by the MM or LLRAM accesses.

2413308
Read of CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2 might return the value from last interrupted write

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, and r0p2. Open.

Description

Cortex-R82 processor implements the Generic Timer registers as defined by the Arm architecture.

Because of this erratum, read of CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2 might return the value from last interrupted write to it.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- Core RAM protection is enabled with IMP_MEMPROTCTLR_EL1.MEMPROTEN = 1
- RAS error reporting is enabled with ERR1CTLR.ED = 1
- One of the ERR<n>STATUS.V, where n is 1 to 3, is changed due to either a new *Error Correcting Code* (ECC) error or a software write
- Software writes to CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2
- An interrupt is taken
- Software reads the last written CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2

Implications

If this erratum occurs, the write might be interrupted so the write is not successful but the read would still return the written value.

If the software exits from the interrupt handler with the preferred return address, the write will be replayed and the read mismatch will disappear.

Workaround

No workaround is expected to be required as the write is expected to be replayed. If the software might switch contexts before returning from the interrupt handler and read one of the CNTKCTL_EL1, CNTHCTL_EL2 or CNTVOFF_EL2, this erratum can be avoided by reading the affected System register and writing the value again.

2416441
Clock is not gated correctly when the core is in the Full retention power mode

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p2. Open.

Description

The Cortex-R82 processor implements per core power modes, including, optionally, Full retention (FULL_RET) which allows the core to save power while retaining the values of its memory and registers.

Because of this erratum, when a core enters the FULL_RET power mode due to a *Wait for Event* (WFE) or *Wait for Interrupt* (WFI) instruction and at the same time an event arrives targeting that core, this can prevent the clock from being gated correctly while the core is in the FULL_RET power mode. Furthermore, some types of events could fail to wake up the core for the WFE case.

Configurations affected

This erratum affects configurations of the Cortex-R82 processor that implement Full Retention for logic retention (auxiliary configuration parameter FULL_RET set to 1).

Conditions

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

1. IMP_CPUPWRCTLR_EL1.WFIRET > 0 or IMP_CPUPWRCTLR_EL1.WFERET > 0
2. A core executes a WFE or WFI instruction
3. The core transitions to the FULL_RET power mode
4. During the transition, or at any time while in the FULL_RET power state, an event arrives from another core or component in the system

Implications

If this erratum occurs, the clock to the core can toggle when the logic is in a retention state. This could cause undesirable physical effects in the power domain, including damaging the device. Additionally, depending on complex microarchitectural timing conditions, data corruption or system deadlock could also occur.

Workaround

To avoid this erratum, do not enable the FULL_RET power mode. This can be achieved by setting IMP_CPUPWRCTLR_EL1.WFIRET and IMP_CPUPWRCTLR_EL1.WFERET to their default value of 0.

Category B (rare)

There are no errata in this category.

Category C

2110865
RAM ECC errors or poison during core power off transitions may lead to deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor has the capability of detecting *Error Correcting Code* (ECC) errors in RAMs and flagging an external *Reliability*, *Availability*, *and Serviceability* (RAS) IRQ interrupt.

The processor can receive poison from external interfaces towards the *LLRAM Coherency Unit* (LCU) and L2 cache through the *Low-latency RAM* (LLRAM) and *Main Master* (MM) interfaces respectively.

The processor can also power down its cores.

If the processor generates an ECC error or poison triggered RAS interrupt during a core power off sequence, the processor deadlocks.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when all the following conditions are met:

- Based on the error type, a relevant field or fields (CI, DUI, CFI, FI, and UI) of the ERR<n>CTLR is set to 1 to generate an IRQ
- Error reporting is enabled (ERR<n>CTLR.ED = 1)
- A request to transition a core to the off power mode is underway
- Either an ECC error occurs in a RAM or poison is received from the LLRAM or MM interfaces

Implications

If this erratum occurs, the processor deadlocks.

Workaround

To avoid this erratum, set the CI, DUI, CFI, FI, and UI fields in the ERR<n>CTLR to 0 before requesting a transition to the off power mode for any core.

2131519
RAS Fault Injection to shared memory may cause deadlock or data corruption

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor supports fault injection for testing of fault handling software. When a *Corrected Error* (CE) or *Deferred Error* (DE) is injected to Error Record 3 (for the shared memories), it may cause a deadlock or data corruption.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when all the following conditions are met:

- A CE or DE fault is requested to be injected to the Eror Record 3
- The fault is injected on the second beat of a transaction

Implications

When this erratum occurs, the L2 memory system will send a spurious response to the L1 memory system which could cause deadlock or data corruption.

This erratum does not affect injection of an *Uncontainable Error* (UC) to the Error Record 3 or any error injection to other Error Records.

Workaround

There is no workaround. It is expected that on test silicon, software can use other Error Records to test fault injection.

2131811
DSB does not guarantee the observability of the effects of the GIC CPU interface register accesses

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements the *Generic Interrupt Controller* (GIC) CPU interface. The GIC architecture requires that the execution of a DSB instruction guarantees the effects of the GIC CPU interface register accesses are observed by the Distributor and Redistributor.

Because of this erratum, execution of a DSB instruction after accessing to the GIC CPU interface registers cannot guarantee the observability of the effects. Under certain conditions, a spurious interrupt might be observed or a new interrupt might be lost.

Configurations affected

The erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when the GIC CPU interface is enabled (**GICCDISABLE** signal LOW at reset) and either of the following set of conditions is met:

1. For the first set of conditions to cause this erratum, the following sequence of conditions must occur:
   a. Program executes a DSB instruction after a write to either of the following:
      - One of the ICC_EOIR0_EL1, ICC_EOIR1_EL1, ICV_EOIR0_EL1, and ICV_EOIR1_EL1 registers when EOImode is not set
      - One of ICC_DIR_EL1 and ICV_DIR_EL1 registers to deactivate an interrupt
   b. Program writes 1 to the GICD_ISACTIVER<n> or the GICR_ISACTIVER0 registers to re-activate the interrupt
2. For the second set of conditions to cause this erratum, the following sequence of conditions must occur:
   a. Program executes a DSB after a read to one of the ICC_IAR0_EL1 and ICC_IAR1_EL1 registers to acknowledge the interrupt
   b. Program clears the source of the interrupt in the peripheral
   c. A new interrupt is asserted by the peripheral

Implications

If the first set of conditions causes this erratum, a spurious interrupt might be observed.

If the second set of conditions causes this erratum, the new interrupt might be lost.

Workaround

No workaround is required to avoid the implications of the first set of conditions. This is because program can tolerate the spurious interrupt in this case.

To avoid the implications of the second set of conditions, replace the DSB with two writes to the ICC_DIR_EL1 register with an unused but supported SPI INTID before clearing the source interrupt in the interrupt handling routine.

2140942
Incorrect value reported in RAS registers on L2 cache ECC errors

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor reports and records *Error Correcting Code* (ECC) errors in memories and system ports according to the *Reliability*, *Availability*, *and Serviceability* (RAS) Extension. Because of this erratum, fields that indicate the primary error code or the error type might be incorrect in the error record registers 3, 4, and 5 that are associated with the L2 cache RAMs (L2 cache data RAMs, L2 cache tag RAMs, L2 cache data buffer RAMs, L2 duplicate L1 tag RAMs).

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when RAM protection is enabled (IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN=1) and one of the following conditions applies:

- Multiple ECC errors (single bit, double bit, or both) occur in different L2 cache RAMs
- A single bit ECC error occurs in the L2 cache data buffer RAM and is deferred

Implications

If this erratum occurs, the information in the ERR<n>MISC0.TYPE and ERR<n>STATUS.SERR fields could be incorrect for the error record register 3, 4 or 5.

There is still significant benefit gained from the ECC logic because this erratum does not impact the detection or correction of the ECC errors. It only affects the reporting of the L2 cache RAM errors in the error record registers.

Workaround

There is no workaround.

2151159
Multi-core Cortex-R82 may deadlock when executing L2DBG operations and WFx events

Status:

Affects: Cortex-R82
Fault type: Programmer Category C
Fault status: r0p0. Fixed in r0p1.

Description:

In a system with the Cortex-R82 processor that has two or more cores, the system can deadlock if multiple cores are doing L2 cache debug operations while simultaneously trying to do a power transition to either OFF or OFF_EMU power modes in a core or there is a change in the L2 cache operating mode (From FULL RAM to DL1ONLY).

Configurations affected:

This erratum affects all configurations of the Cortex-R82 processor configured with **NUM_CORES** >= 2

Conditions:

This erratum occurs when there is a core transitioning to OFF or OFF_EMU power mode or the L2 cache operating mode changes from FULL RAM to DL1ONLY and when all the following conditions are met:

- One of the SYS IMP_CLUSTERCDBGL2D, SYS IMP_CLUSTERCDBGL2DT, or SYS IMP_CLUSTERCDBGL2T is executed in two cores
- At least one of the operations above is followed by a WFI or WFE request without an IMP_CLUSTERCDBGDR0_EL1 read

Implications:

If this erratum occurs, the processor deadlocks.

It is expected that the erratum conditions do not represent plausible software. This is because of the absence of an IMP_CLUSTERCDBGDR0_EL1 read before WFI/WFE, and the implicit requirement to execute in the highest Exception level.

Workaround:

The problematic system instructions can be trapped to EL2. To avoid this erratum, always perform a System register read of IMP_CLUSTERCDBGDR0_EL1 following the execution of SYS IMP_CLUSTERCDBGL2D, SYS IMP_CLUSTERCDBGL2DT, or SYS IMP_CLUSTERCDBGL2T before issuing a WFI or WFE request.

2169550
ETM trace might not report certain direct branch instructions

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements the *Embedded Trace Macrocell* (ETM). The ETM generates the trace packets of instructions.

Because of this erratum, the ETM might not generate atom elements for certain direct branch instructions under very specific microarchitectural conditions.

Configurations affected

The erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when all the following conditions are met:

- ETM trace is enabled and not prohibited
- The core executes a direct branch instruction which is triple-issued with a load instruction and a store instruction but the direct branch instruction is not the last of the three instruction
- An interrupt is pending and not masked
- One of the load/store instructions and the direct branch instruction are completed
- The other load/store instruction has been stalling and is subsequently interrupted

Implications

If this erratum occurs, the ETM does not generate:

- An Atom element for the branch instruction even though it was executed
- An Address element indicating the target of the branch instruction

The preferred exception return address provided with the Exception element correctly indicates where the exception is taken from. This might cause the trace analyzer to infer incorrect instruction execution.

Workaround

A workaround is not expected to be required for test silicon. This is because the erratum conditions are rarely met.

A trace analyzer might be able to detect that this erratum has occurred:

- If a branch instruction is encountered when inferring execution due to an Exception element
- When load or store instructions are P0 instructions, a load or store instruction is encountered when inferring execution due to an Exception element
- If the preferred exception return address for an Exception element indicates the exception was taken from an address which is at a lower address than the target of the previous P0 element

If a workaround is required, the erratum can be avoided by setting IMP_CPUACTLR_EL1.LSDI to 0 to disable dual-issuing load/store instructions. This has high performance impact.

2170009
Error injection via ERR0PFGCTL might occur at the wrong time

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor supports fault injection for testing of fault handling software. When a *Corrected Error* (CE), *Deferred Error* (DE), or *Uncontainable Error* (UC) is injected to the Error Record 0 (for the per-core memories), the injection might occur at an unexpected time. If fault injection is configured to restart whenever the counter (ERR0PFGCDN_EL1) reaches zero, fault injection may be at a higher rate than expected.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all of the following conditions are met:

- Error detection is enabled (IMP_MEMPROTCTLR_EL1.MEMPROTEN = 1)
- Fault injection is requested (ERR0PFGCTL.CDNEN = 1 and one of ERR0PFGCTL.{CE, DE, UC} is set to 1)
- An access that would cause fault injection has previously occurred

Implications

If this erratum occurs, fault injections to the Error Record 0 might take place sooner than expected. Errors might be injected whenever the counter reaches zero, without requiring a triggering access as long as one has previously occurred. As a result, ERR0MISC0 might contain invalid information. If fault injection is configured to restart whenever the counter reaches zero, the fault injection rate may be higher than expected.

Workaround

There is no workaround. It is expected that on test silicon, software may be able to cope with the fault being injected sooner and not rely on the information in ERR0MISC0.

2173953
PMU event counts might be inaccurate

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements a number of performance monitor events. Because of this erratum, some of the events will not count correctly.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when one of the *Performance Monitoring Unit* (PMU) event counters is configured to count one of the following events:

- 0x0001 L1I_CACHE_REFILL
- 0x000F UNALIGNED_LDST_RETIRED
- 0x0010 BR_MIS_PRED
- 0x0012 BR_PRED
- 0x0015 L1D_CACHE_WB
- 0x0020 L2D_CACHE_ALLOCATE
- 0x0043 L1D_CACHE_REFILL_WR
- 0x0070 LD_SPEC
- 0x0071 ST_SPEC
- 0x0072 LDST_SPEC
- 0x0076 PC_WRITE_SPEC
- 0x00C1 L2D_CACHE_REFILL_PREFETCH
- 0x0331 TCMS_ACCESS_WR
- 0x0361 LLRAM_L1D_CACHE_REFILL_WR
- 0x0362 LLRAM_L1D_CACHE_REFILL

Implications

If this erratum occurs:

- Any count of UNALIGNED_LDST_RETIRED, BR_MIS_PRED, BR_PRED, L2D_CACHE_ALLOCATE,

L1D_CACHE_REFILL_WR, LD_SPEC, ST_SPEC, LDST_SPEC, PC_WRITE_SPEC, TCMS_ACCESS_WR or L2D_CACHE_REFILL_PREFETCH might be lower than it should be.
- Any count of L1D_CACHE_WB or L1I_CACHE_REFILL might be higher than it should be.
- Any count of LLRAM_L1D_CACHE_REFILL_WR or LLRAM_L1D_CACHE_REFILL would be 0. Software will not be able to use these events for performance analysis.

Workaround

There is no workaround.

2215960
An atomic with Acquire semantics may be observed before an earlier Store-Release to TCM or LLPP

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor contains *Tightly Coupled Memories* (TCMs) and a *Low-latency Peripheral Port* (LLPP) that are private to the core. Due to this erratum, an atomic instruction with Acquire semantics could be observed by a different observer before a store with Release semantics performed to the TCMs or LLPP that was earlier in program order.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

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

1. There is a Store-Release instruction to the TCMs or LLPP
2. There is an atomic instruction with Acquire but not Release semantics to a different memory port than the Store-Release

Implications

When this erratum occurs, the memory effects of the atomic with Acquire semantics could be observed by a different agent before the memory effects of the Store-Release. For the effects of this erratum to be observed, the Store-Release must be visible to another core, as in cases where the value is read from the TCMs through the *ACE-Lite Slave* (ACELS) port or where the LLPP is connected to a shared interconnect.

For the memory that is intended to be shared between cores, it is recommended to use the *Main Memory* (MM) port or *Low-latency RAM* (LLRAM) port, or for shared peripherals the *Shared Peripheral Port* (SPP).

Workaround

To avoid this erratum, a DMB ST can be inserted between the Store-Release and the atomic with Acquire semantics.

2218751
Data cache maintenance operations by Set/Way targeting a 4MB L2 cache might affect incorrect cache index

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor supports an L2 cache that can be configured to a maximum size of 4MB. When the cache size is configured as 4MB, the Set/Way cache maintenance operations (DC CISW, DC CSW, DC ISW) could affect the wrong index when targeting the L2 cache and when the targeted index has the most significant bit equal to 1.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor where the L2 cache size is 4MB (**L2_CACHE_SIZE** = 4096).

Conditions

This erratum occurs when the following conditions are met:

- A Set/Way cache maintenance operation is executed targeting the L2 cache
- The most significant bit of the Set is 1. For example, bit [18] of the argument to the DC instruction is 1

Implications

When this erratum occurs, software will be unable to perform Set/Way data cache maintenance operations to the L2 cache indexes with the most significant bit set. No data corruption can occur due to this erratum but data may not be externally visible to other agents after a Set/Way operation.

The use of Set/Way instructions to manage coherency is discouraged because the memory accesses from other cores could cause the line to be migrated to the cache of another core. Because of this, typical software is not expected to use Set/Way cache maintenance and as such the implications of this erratum are not expected to be significant.

The cache invalidation performed as part of the power down sequence or after reset is unaffected by this erratum.

Workaround

There is no workaround.

2224832
Debug APB accesses to reserved core address is accessible when core powered off

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a read or write transaction is made on the Debug APB interface to a reserved address within the core address space, then it should be treated as RAZ/WI while the core is powered on. If certain reserved addresses are accessed, then they will be treated as RAZ/WI even when the core is powered off, instead of giving an error response.

Configurations affected

This erratum affects all configurations of the Cortex-R82.

Conditions

The erratum occurs when a read or write access is made on the Debug APB interface that is to an address 0x00nm_0D90 where n is from 8 to (8+NUM_CORES-1) and m is from 0 to 4.

Implications

Software is not expected to access these reserved registers, therefore there are no implications.

Workaround

No workaround is necessary.

2226045
ATB flush is unreliable

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description:

The Cortex-R82 processor provides two ATB buses for trace from the internal *Embedded Trace Macrocells* (ETMs) and *Embedded Logic Analyzers* (ELAs). These buses support an external flush request which should be used to ensure the final packets of trace from a session have been captured. Due to this erratum, there are scenarios where the processor will acknowledge a flush request but will not output all of the trace packets or may output trace packets after the flush is acknowledged.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

The erratum occurs if the following sequence of conditions is met:

1. The ETM or ELA is enabled
2. Trace data is generated on the ATB bus
3. An external flush request is generated

Implications

If this erratum occurs, the final few bytes of a trace stream which are expected may not be captured by the Trace Capture Device, and some of the trace packets from the data trace stream may not be output until another external flush request is received. It is not possible to determine if a flush request has been successful.

Due to this erratum, in a system with additional downstream trace infrastructure, trace will not reliably drain from some components (such as upsizers).

Workaround

There is no workaround to reliably avoid this erratum. Generating several flush requests before using a flush and stop sequence is the most reliable way to drain trace data from the processor.

2246499
Some RAS and Debug identification registers have wrong values

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements CoreSight registers to identify the components.

Because of this erratum, read of EDDEVARCH.ARCHPART, ERRDEVARCH.ARCHPART, ERRCIDR1, or ERRDEVAFF returns wrong values.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when software reads EDDEVARCH, ERRDEVARCH, ERRCIDR1, or ERRDEVAFF registers.

Implications

If this erratum occurs:

- Read of EDDEVARCH.ARCHPART returns 0xA15 (A Profile Debug) instead of the correct value 0xA05 (R Profile Debug)
- Read of ERRDEVARCH.ARCHPART returns 0x500 instead of the correct value 0xA00 (RAS)
- Read of ERRCIDR1 returns 0x90 instead of the correct value 0xF0
- Read of ERRDEVAFF returns the value of MPIDR_EL1 instead of the correct value 0x0

Workaround

There is no workaround.

2248150
Accesses to Non-secure EL2 physical timer registers and Secure EL2 virtual timer registers are not UNDEFINED

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements Generic Timer registers as defined in the Arm architecture. According to the architecture, Non-secure EL2 physical timer registers and Secure EL2 virtual timer registers are not implemented and therefore, they are **UNDEFINED**.

Because of this erratum, accesses to the Non-secure EL2 physical timer registers and Secure EL2 virtual timer registers are not **UNDEFINED**.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

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

- Software reads or writes CNTHP_CVAL_EL2, CNTHP_TVAL_EL2, or CNTHP_CTL_EL2 in EL2
- Software reads or writes CNTHVS_CVAL_EL2, CNTHVS_TVAL_EL2, or CNTHVS_CTL_EL2 in EL2

Implications

If this erratum occurs:

- The accesses to the Non-secure EL2 physical timer registers and Secure EL2 virtual timer registers are not **UNDEFINED**
- There are no effects of triggering events from using the Non-secure EL2 physical timer registers
- **nCNTHVSIRQ** can still be asserted LOW when the timer condition for the Secure EL2 virtual timer is met

Workaround

There is no workaround.

2254257
Read of ERRGSR might return incorrect value

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 processor implements the *Reliability*, *Availability*, *and Serviceability* (RAS) Extension as defined in the Arm architecture. The Cortex-R82 processor implements 6 error records. It also implements the ERRGSR which shows the status for the records in the group.

Because of this erratum, the read of ERRGSR might return incorrect value.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

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

1. Core RAM protection is enabled with IMP_MEMPROTCTLR_EL1.MEMPROTEN = 1
2. RAS error reporting is enabled with ERR0CTLR.ED = 1
3. One of the ERR<n>STATUS.V, where n is 0 to 2, is changed due to either a new *Error Correcting Code* (ECC) error or a software write
4. Immediately after that, a different ERR<n>STATUS.V, where n is 0 to 2, is set due to a new ECC error
5. Software reads ERRGSR

Implications

If this erratum occurs, the software might read incorrect value from the ERRGSR.

Workaround

To avoid this erratum, software should read ERR<n>STATUS.V directly for error records 0 to 2 rather than using ERRGSR.

2261658
A single bit hard error on poisoned TCM data could cause a deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The Cortex-R82 has *Error Correcting Code* (ECC) error detection on the *Tightly Coupled Memories* (TCMs). *Single Error Correction*, *Double Error Detection* (SECDED) is supported, as well as the use of poison to defer errors. There is also support to make forward progress in the presence of a single hard error. Due to this erratum, when a hard error occurs affecting a single bit within the same 128 bits as data that has been poisoned in the TCMs, it could cause the processor to deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1) and and a core in the cluster configured with a TCM (**ITCM_SIZE<m>** or **DTCM_SIZE<m>** parameter greater than 0).

Conditions

This erratum occurs when all the following conditions are met:

- RAM protection is enabled (**IMP_MEMPROTCTLR_EL1.MEMPROTEN=1**)
- There is data in one of the TCM memories which is poisoned
- A single bit error occurs in a different ECC granule to the poisoned data, but within the same 128 bits
  - This single bit error is a hard error and so persists after the core attempts to correct it
- A load to the TCM accesses the data with the single bit error, but not the poisoned data

Implications

If this erratum occurs, the processor deadlocks.

There is no impact to error detection mechanisms due to this erratum. However due to this erratum, forward progress is not always guaranteed in the presence of a hard error in the TCMs.

Because this erratum occurs only when there is an ECC error and because ECC errors are not expected in sample silicon, it is expected that this erratum will have no implications for sample silicon.

Workaround

There is no workaround for this erratum.

2284986
PMU event counts might be inaccurate

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements a number of performance monitor events. Because of this erratum, some of the events will not count correctly.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when one of the *Performance Monitoring Unit* (PMU) event counters is configured to count one of the following events:

- Core PMU events:
  - 0x003D STALL_SLOT_BACKEND
  - 0x003E STALL_SLOT_FRONTEND
  - 0x0074 ASE_SPEC
  - 0x0075 VFP_SPEC
  - 0x00E4 STALL_BACKEND_ILOCK
  - 0x00E6 STALL_BACKEND_ILOCK_FPU
  - 0x0327 MM_ACCESS
  - 0x032A LLRAM_ACCESS
  - 0x032D SPP_ACCESS
  - 0x0332 TCMS_ACCESS
  - 0x0378 BARRIER_STB_FULL
  - 0x0379 L1I_WT_HIT
  - 0x0019 BUS_ACCESS
  - 0x0060 BUS_ACCESS_RD
  - 0x0061 BUS_ACCESS_WR
- Cluster PMU events:
  - 0x0160 MACP_ACCESS_RD
  - 0x0161 MACP_ACCESS_WR
  - 0x041A MM_ACTIVE
  - 0x041D MM_CYCLES
  - 0x0461 MM_ACCESS_WR

- 0x0462 MM_ACCESS_SHARED
- 0x0463 MM_ACCESS_NOT_SHARED
- 0x0464 MM_ACCESS_NORMAL
- 0x0465 MM_ACCESS_PERIPH

Implications

If this erratum occurs:

- Any count of STALL_SLOT_BACKEND, VFP_SPEC, STALL_BACKEND_ILOCK, STALL_BACKEND_ILOCK_FPU, MM_ACCESS, LLRAM_ACCESS, SPP_ACCESS, TCMS_ACCESS, MACP_ACCESS_RD, MACP_ACCESS_WR, MM_ACTIVE, MM_CYCLES, MM_ACCESS_WR, MM_ACCESS_SHARED, MM_ACCESS_NOT_SHARED, MM_ACCESS_NORMAL, MM_ACCESS_PERIPH, BUS_ACCESS, BUS_ACCESS_RD or BUS_ACCESS_WR might be lower than it should be
- Any count of STALL_SLOT_FRONTEND, ASE_SPEC or BARRIER_STB_FULL might be higher than it should be
- Any count of L1I_WT_HIT would be 0. Software will not be able to use these events for performance analysis

Workaround

There is no workaround.

2286314
Access to TRCSEQEVR3 is not UNDEFINED

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements the system instruction interface to the *Embedded Trace Macrocell* (ETM) as defined in the Arm architecture. There are 3 registers defined for the TRCSEQEVR<n> so n is in 0 to 2.

Because of this erratum, the access to the TRCSEQEVR3 is not **UNDEFINED**.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

This erratum occurs when software accesses to the TRCSEQEVR3 register.

Implications

If this erratum occurs, software access to the TRCSEQEVR3 is not **UNDEFINED** and the access is RAZ/WI.

Workaround

There is no workaround.

2313135
Cacheable accesses to LLRAM might deadlock in the presence of an ECC error

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor has a *Low-latency RAM* (LLRAM) port to provide a more deterministic shared memory. The LLRAM port is designed on the assumption that it will be connected directly to a low-latency external memory.

If the LLRAM port is not directly connected to a memory but instead to an interconnect which reorders read responses then because of this erratum, and while in the presence of an ECC error, the processor might deadlock.

Configurations affected

This erratum affects configurations in which all the following are true:

- The processor is configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1)
- 4 or less cores are in the cluster (**NUM_CORES** configuration parameter is 4 or less)
- The LLRAM port is not directly connected to an external memory

Conditions

This erratum occurs when all the following conditions apply:

- The LLRAM AXI port is connected to a subordinate that has a read acceptance capacity of 3 or more
- Cacheable accesses are performed to the LLRAM, either explicit accesses or by the prefetcher
  - Read responses to these accesses are reordered
- One of the following occur:
  - A double bit ECC error occurs in the LCU duplicate L1 tag RAM with IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN=1
  - An ECC error in the L1 data cache RAMs occurs with IMP_MEMPROTCTLR_EL1.MEMPROTEN=1
- Complex micro-architectural conditions occur

Implications

If this erratum occurs, the processor deadlocks. This erratum is not expected to have significant implications for most systems as it is expected that the LLRAM port is directly connected to a memory and so read responses will not be reordered. If read responses are not reordered, then the conditions for this erratum cannot occur.

Workaround

It is not expected that a workaround is required for this erratum. However, if a system is susceptible to this erratum and a workaround is required, disable the RAM protection by setting IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN to 0 and IMP_MEMPROTCTLR_EL1.MEMPROTEN to 0.

2317561
Halting step syndrome might be wrong on stepping a Load-Exclusive instruction

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

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-R82 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.

2330237
Multi-issuing control with IMP_CPUACTLR_EL1.MI = 0b010 is wrong

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Cortex-R82 processor implements multi-issuing controls with IMP_CPUACTLR_EL1.MI. When IMP_CPUACTLR_EL1.MI is programmed as 0b010, FP/AdvSIMD instructions can only be issued from slot0 while other instructions can be issued from slot0 or slot1.

Because of this erratum, other instructions can be issued from slot2 not respecting the control of IMP_CPUACTLR_EL1.MI.

Configurations affected

This erratum only affects configurations of the Cortex-R82 processor that have the Advanced SIMD and floating-point support enabled.

Conditions

This erratum occurs when the following condition is met:

- Software programs IMP_CPUACTLR_EL1.MI as 0b010

Implications

If this erratum occurs, FP/AdvSIMD instructions can still only be issued from slot0 but other instructions can be issued from slot0, slot1 or slot2.

Workaround

There is no workaround.

2338818
IFU Cache errors might report incorrect BANK to the RAS node

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor reports and records *Error Correcting Code* (ECC) errors in memories and system ports according to the *Reliability*, *Availability*, *and Serviceability* (RAS) Extension. Because of this erratum, fields that indicate the bank of the error may be reported inconsistently for Instruction cache data RAMs compared to other RAMs. This applies to the error record registers 1, 2, and 3 that are associated with each core.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include RAM protection (**RAM_PROTECTION** configuration parameter is set to 1).

Conditions

This erratum occurs when all the following conditions are met:

- RAM protection is enabled (**IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN=1**)
- An ECC error occurs in one of the Instruction cache data RAMs.

Implications

If this erratum occurs, the bank of the error reported will be indicated by the value **ERR<n>MISC0.BANK** and the subbank in **ERR<n>MISC0.CHUNK** fields, rather than both being reported in the **ERR<n>MISC0.BANK** field as for other RAMs.

Workaround

There is no workaround.

2339884
Cluster CTI trigger outputs are not reaching cluster ELA CTITRIGINs

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0. r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor can optionally implement an *Embedded Logic Analyzer* (ELA) at the cluster level. This ELA has trigger inputs from the cluster *Cross Trigger Interface* CTI.

Because of this erratum, the ELA will never observe triggers generated from the cluster CTI.

Configurations affected

This erratum affects configurations of the Cortex-R82 processor with embedded logic analyzers included.

Conditions

This erratum occurs when a trigger from the cluster CTI intended for the cluster ELA is generated.

Implications

The cluster ELA cannot be triggered from events observed by the cluster CTI. There are no alternative trigger inputs to this ELA.

Workaround

There is no workaround to this erratum.

2358852
Pointer authentication key registers are not initialized

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Cortex-R82 is required to initialize all programmer-visible registers to known values.
Because of this erratum, the pointer authentication key registers are not initialized.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

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

1. Core comes out of warm or cold reset which is not because of debug recovery.
2. Software reads the pointer authentication key registers.

Implications

If this erratum occurs, the read will return uninitialized values which might be from the last written values before the reset.

Workaround

This erratum is not expected to require a workaround.

2359684
Some unallocated debug and trace System registers might be trapped by HCR_EL2.TIDCP

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

In Cortex-R82, 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-R82 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.

2370219
Use of debug related power states can result in deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor can use certain power states to allow debug of power transition sequences and fault conditions.

Because of this erratum, if an interrupt is received during any core transitioning between specific debug related power states, the processor can deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following sequence of conditions is met for a single core:

1. A core is required to wake up through either of the following:
   - The core was entering OFF mode from WFI, but encountered an *Error Correcting Code* (ECC) error signaling a *Reliability, Availability, and Serviceability* (RAS) interrupt that aborted the power transition
   - The core is transitioning OFF to ON, because its COREWAKEREQUEST pin was asserted by the *Generic Interrupt Controller* (GIC)
2. A debug-related power transition is requested for the cluster through one of the following:
   - The cluster transitions ON to WARM_RST or DBG_RECOV
   - The cluster transitions OFF_EMU to OFF
   - The cluster transitions MEM_RET_EMU to MEM_RET

Implications

If this erratum occurs, outstanding power transitions could cause the processor to deadlock. These transitions are only expected to occur during debug of a device.

Workaround

To avoid this erratum, ensure that the following debug-related cluster power transitions do not occur:

- Avoid transitions OFF_EMU to OFF and MEM_RET_EMU to MEM_RET
- Avoid using WARM_RST and DBG_RECOV to debug a RAS error

2375011
PSTATE.SS not restored correctly if OSLK is set to 1 just before debug exit

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements debug features up to v8.4 and the optional debug over powerdown feature from v8.5 (FEAT_DoPD). According to the architecture, the debug exit is a context synchronization event which uses the newly synchronized state to perform the PE state restoration.

Because of this erratum, the restoration of PSTATE.SS will be calculated based on old state. Under certain conditions, the PSTATE.SS will have the wrong value after the debug exit.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

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

- The Core is in debug state with DSPSR_EL0.SS=1 and OSLK=0.
- An MSR OSLAR_EL1 is executed, setting OSLK=1.
- A debug exit is performed, triggering the PSTATE.SS to be restored from the DSPSR.SS.

Implications

If this erratum occurs, PSTATE.SS will be set after exiting debug state, instead of being masked to zero.

Workaround

An ISB can be executed after the MSR to synchronize the OSLK value, before performing the debug exit.

2384696
DPU returns wrong value on MRS to DLR_EL0

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements debug features up to Armv8.4 and the optional debug over powerdown feature from Armv8.3 (FEAT_DoPD). According to the architecture, the DLR_EL0 register can be accessed from debug state using an MRS.

Because of this erratum, the read value from the DLR will always be word-aligned and might be different than the actual value.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor.

Conditions

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

- The core is in debug state
- DLR_EL0 contains a PC value that is not word-aligned (bits [1:0] != 00)
- An MRS to the DLR_EL0 is executed

Implications

If this erratum occurs, the MRS will return the DLR value word-aligned ([1:0] == 0)

Workaround

There is no workaround.

2386524
Defective RAS registers

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1. Fixed in r0p2.

Description

The Cortex-R82 processor implements the *Reliability, Availability and Serviceability* (RAS) extension, up to and including the ARMv8.4-RAS.

Because of this erratum, accesses to ERR4PFGCDN, ERR1PFGCTL, and ERR4PFGCTL might be impacted.

Configurations affected

This erratum affects any configuration of Cortex-R82 processor that includes RAM protection (RAM_PROTECTION configuration parameter is set to 1).

Conditions

For the erratum part concerning the ERR4PFGCDN to occur, the following condition applies:

- A read or write operation is performed via the Utility Bus

For the erratum part concerning the ERR1PFGCTL and ERR4PFGCTL to occur, the following condition applies:

- A read operation is performed via the Utility Bus or by an MRS ERXPFGCTL

Implications

When this erratum occurs:

Implications for ERR4PFGCDN:

- A write access performed via the Utility Bus will have no effect
- A read access performed via the Utility Bus will always return 0

Implications for ERR1PFGCTL and ERR4PFGCTL:

- Bit 12 of these registers will be read as 0b0, instead of the expected 0b1

Workaround

To avoid this erratum for ERR4PFGCDN:

- Software running on the core can access the ERR4PFGCDN by MRS/MSR ERXPFCDN_EL1

To avoid this erratum for ERR1PFGCTL and ERR4PFGCTL:

- Software can ignore the read value for bit 12 and assume it has returned a value of 0b1

2416442
Multiple double bit ECC errors on core powerdown might deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p2. Open

Description

The Cortex-R82 processor contains logic to clean and invalidate caches when a core is powered down. If multiple double bit *Error Correcting Code* (ECC) errors occur when a core is being powered down, then it could cause the core to deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-R82 process with an L2 cache (**L2_CACHE_SIZE** > 0) and configured to include RAM protection (**RAM_PROTECTION** == 1).

Conditions

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

1. A core is being powered down.
2. A double bit ECC error occurs in a L2DB while it contains data evicted from the L1 data cache as part of the power down sequence.
3. Another double bit ECC error occurs in a different L2DB containing data evicted from the L1 data cache for a different way, but for the same set.
4. Requests from other cores are performed to the same index as the data held in the L2DBs which have double bit ECC errors, during the short time window before the data is written out on the bus.

Implications

If this erratum occurs, the Cortex-R82 processor deadlocks.

Workaround

No workaround is expected to be needed due to the rarity of multiple double-bit errors in close proximity. If a workaround is required, the L1 data cache can be manually invalidated with set/way operations before powering down a core.