Arm Cortex-R82 Processor MP130

Software Developer Errata Notice

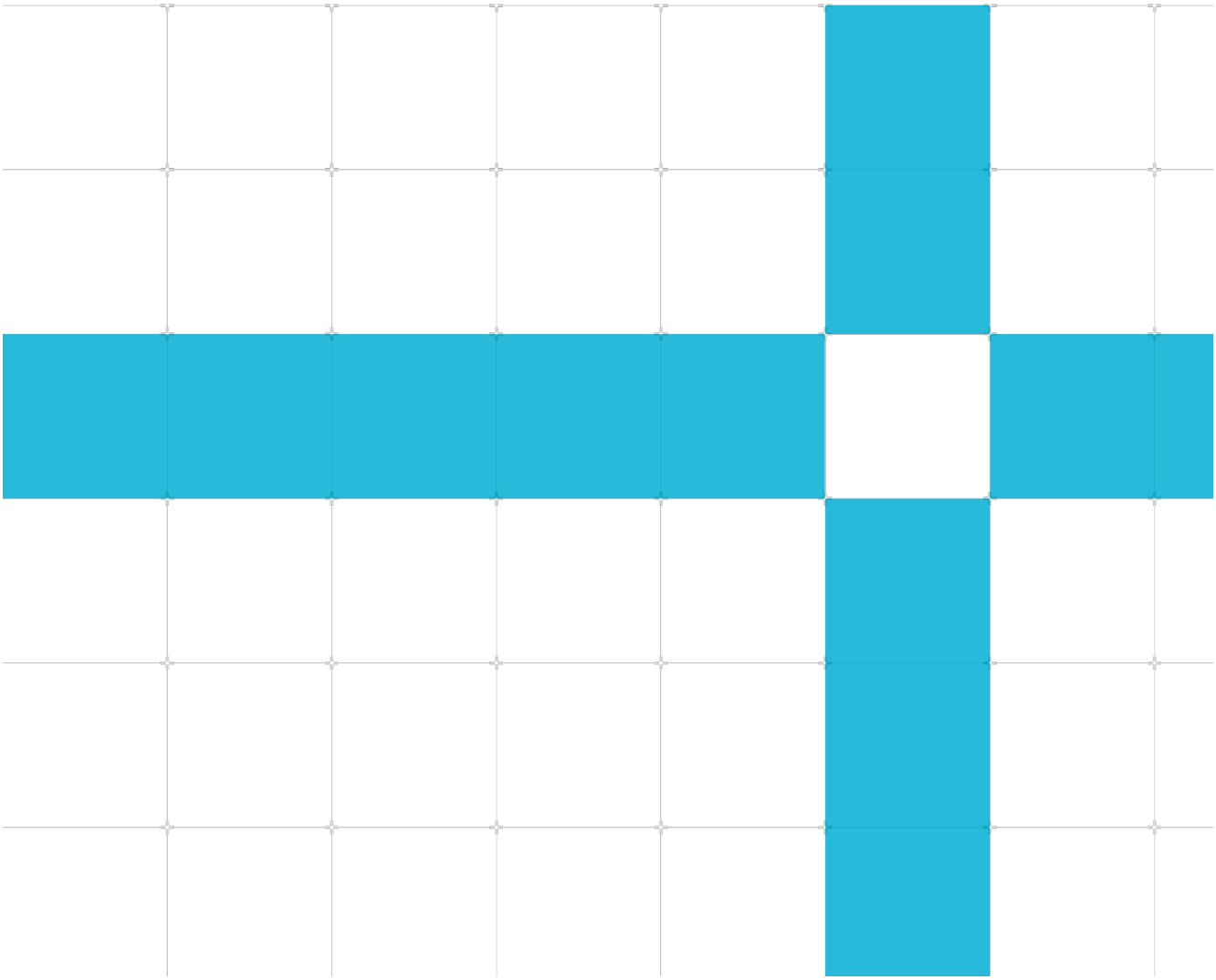
Date of issue: February 28, 2025

Non-Confidential

Document version: 17.0

Document ID: SDEN-1956900

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



This document (SDEN_1956900_17.0_en) was issued on February 28, 2025.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-1956900**

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**.

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.

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.

February 28, 2025: Changes in document version v17.0

ID	Status	Area	Category	Summary
3823246	New	Programmer	Category C	Cluster PMU does not generate overflow

October 31, 2024: Changes in document version v16.0

ID	Status	Area	Category	Summary
3114083	Updated	Programmer	Category C	Cluster OFF_EMU to ON transition stuck because PACCEPT is blocked due to debug requests during OFF_EMU
3683909	New	Programmer	Category C	The CPU ROM Table reports an incorrect base address for CTI2-7 for configurations with DENSE_CS_ADDR_MAP = 1
3686657	New	Programmer	Category C	Interrupt might be taken on a read from device memory that receives a bus error
3769700	New	Programmer	Category C	L2 incorrectly applies address-based hazarding to Non-Reorderable Device writes, reducing performance
3776502	New	Programmer	Category C	L2 incorrectly applies address-based hazarding to Reorderable Device writes, reducing performance

August 14, 2024: Changes in document version v15.0

ID	Status	Area	Category	Summary
3195930	New	Programmer	Category B	CHI atomic load/cmp/swap with NDERR response and ACP read to same address can lead to deadlock
3394103	New	Programmer	Category B	Precise ECC error in L1 D-cache tag RAM when using both MM and LLRAM ports might lead to data corruption
3338838	New	Programmer	Category C	IMP_CLUSTERACTLR_EL1.L2SPEC specification is incorrect
3378740	New	Programmer	Category C	ECC error in L1 D-cache tag RAM might not get logged in RAS record
3454202	New	Programmer	Category C	Unused utility bus address locations are aliased
3545298	New	Programmer	Category C	An external Non-cacheable store exclusive receiving a bus error response might not raise an asynchronous abort
3606619	New	Programmer	Category C	L2 Flushes too many ways when moving from FULL_RAM to HALF_RAM
3639153	New	Programmer	Category C	Cold debug recovery reset will not reset some system registers

March 25, 2024: Changes in document version v14.0

ID	Status	Area	Category	Summary
3299478	New	Programmer	Category B	DSB SY might deadlock when executed after a context switch
3321568	New	Programmer	Category B	Execution in VMSA might deadlock after entering WFx
3114083	New	Programmer	Category C	Cluster OFF_EMU to ON transition stuck because PACCEPT is blocked due to debug requests during OFF_EMU
3196425	New	Programmer	Category C	Incorrect VA[14:12] sent to Instruction Cache for SnpDVMOp Virtual I-Cache Invalidate from CHI
3197529	New	Programmer	Category C	AT operation might incorrectly indicate a Synchronous External abort in PAR_EL1
3203547	New	Programmer	Category C	IMP_INTLATENCY_EL2.DEV might cause a Synchronous External abort for IC instructions to Device addresses

December 11, 2023: Changes in document version v13.0

ID	Status	Area	Category	Summary
3100694	New	Programmer	Category B	Defined interrupt latency bound might be violated
3088388	New	Programmer	Category B (rare)	Cluster configuration registers can be corrupted after leaving memory retention state
3105480	New	Programmer	Category C	FAR register should record the lowest address that caused a fault
3123474	New	Programmer	Category C	Some PMU events count incorrectly

October 25, 2023: Changes in document version v12.0

ID	Status	Area	Category	Summary
2753598	New	Programmer	Category B	Deferred error might become uncontainable
3008524	New	Programmer	Category B	ETM4 not accessible using DENSE_CS_ADDR_MAP configuration
3014884	New	Programmer	Category B	Activation of interrupt might not be delivered to GIC
2224832	Updated	Programmer	Category C	Debug APB accesses to reserved core address is accessible when core powered off
2849849	New	Programmer	Category C	An uncontainable error might prevent MM port from making forward progress
2929491	New	Programmer	Category C	Dynamic OP_MODE transition can generate two transitions
3007301	New	Programmer	Category C	Auxiliary Fault Status Registers (AFSR0_ELx) can report wrong values corresponding to some Exception Syndrome Registers (ESR_ELx) faults
3025593	New	Programmer	Category C	Unaligned exclusive to Normal Non-cacheable memory is not aborted

April 04, 2023: Changes in document version v11.0

ID	Status	Area	Category	Summary
2780544	New	Programmer	Category C	Reset of register fields with UNKNOWN values does not work as expected in the presence of Debug recovery mode
2798198	New	Programmer	Category C	Incorrect system reset behavior under MEM_RET and MEM_RET_EMU modes for IMP_CLUSTERACTLR_EL1.UBACCLVL
2842855	New	Programmer	Category C	AFSR0_ELx might have stale value after taking an SError interrupt
2847045	New	Programmer	Category C	External read of TCM while CPUHALT<m> is HIGH might see unexpected ECC errors

November 18, 2022: Changes in document version v10.0

ID	Status	Area	Category	Summary
2725097	New	Programmer	Category B	Interrupting a DSB might cause the DSB for earlier TLBI/IC maintenance operations to fail
2772412	New	Programmer	Category B	DFB does not wait for completion of IC/TLBI from other contexts
2776909	New	Programmer	Category B	A write to LLRAM following a hardware access flag update could result in incorrect translation
2795835	New	Programmer	Category B	A translation table walk may raise an unexpected abort
2652888	Updated	Programmer	Category C	Receiving poison on invalid data lanes might result in a data abort when poison is not supported
2719885	New	Programmer	Category C	Software step exception might set the ESR_ELx.ISV to a wrong value
2729228	New	Programmer	Category C	Debug accesses via APB are not supported in EMU power states
2729398	New	Programmer	Category C	L2RDLAT and L2WRLAT fields in IMP_CLUSTERACTLR_EL1 are hardcoded
2747728	New	Programmer	Category C	Some cluster PMU events are incorrect
2753234	New	Programmer	Category C	An external atomic receiving poison might not raise an asynchronous abort
2763783	New	Programmer	Category C	Speculative accesses might still be made when LLRAM is disabled
2773193	New	Programmer	Category C	32-bit accesses to ERR<n>MISC0 (n= 1-6) will not work as expected
2788717	New	Programmer	Category C	An interrupted atomic operation with bus error to LLRAM may raise unexpected asynchronous abort

July 19, 2022: Changes in document version v9.0

ID	Status	Area	Category	Summary
2639772	New	Programmer	Category B	Turning a core ON while the cluster is being powered OFF may deadlock
2666600	New	Programmer	Category B	CLUSTERPWRDN.PWRDWN is not functional
2684598	New	Programmer	Category B	Software step might go to wrong state
2691481	New	Programmer	Category B	Memory accesses can use old values of some of the Generic System control registers
2604629	New	Programmer	Category C	ETM data trace values for ST atomics are incorrect
2617173	New	Programmer	Category C	An external atomic receiving a SLVERR or DECERR might not raise an asynchronous abort
2652888	New	Programmer	Category C	Receiving poison on invalid data lanes might result in a data abort when poison is not supported
2656238	New	Programmer	Category C	New interrupts might not be taken after an atomic instruction is interrupted
2659537	New	Programmer	Category C	Spurious ECC error might be reported when changing operating mode from DL1ONLY to FULL RAM
2661859	New	Programmer	Category C	Incorrect value reported in RAS registers on LLRAM receiving poison when ECC is not supported
2693781	New	Programmer	Category C	Quality of Service is applied to Main Manager accesses when disabled
2694966	New	Programmer	Category C	Fault handling interrupt can cause power transition to deadlock

April 22, 2022: Changes in document version v8.0

ID	Status	Area	Category	Summary
2455174	New	Programmer	Category B	Cluster MEM_RET is not sustained
2487068	New	Programmer	Category B	Cluster IRQs may be lost in the presence of cluster MEM_RET and OFF power transitions
2440118	New	Programmer	Category C	Some PMU events count incorrectly
2445341	New	Programmer	Category C	Cache initialization and reset of register fields with UNKNOWN values do not work as expected after Debug recovery
2618272	New	Programmer	Category C	When toggling SPIDEN or DBGEN pins close to a core reset, the system might deadlock

February 21, 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

December 13, 2021: Changes in document version v6.0

ID	Status	Area	Category	Summary
2275980	New	Programmer	Category B	Cluster transitions to a lower power mode might 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

November 12, 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

September 23, 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

August 19, 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

July 27, 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

March 26, 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 might cause deadlocks in the presence of Core transitions to higher power modes	r0p0, r0p1, r0p2, r1p0, r1p1	r2p0
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

ID	Area	Category	Summary	Found in versions	Fixed in version
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	r1p0
2416441	Programmer	Category B	Clock is not gated correctly when the core is in the Full retention power mode	r0p2	r1p0
2455174	Programmer	Category B	Cluster MEM_RET is not sustained	r0p0, r0p1, r0p2	r1p0
2487068	Programmer	Category B	Cluster IRQs may be lost in the presence of cluster MEM_RET and OFF power transitions	r0p0, r0p1, r0p2	r1p0
2639772	Programmer	Category B	Turning a core ON while the cluster is being powered OFF may deadlock	r0p0, r0p1, r0p2	r1p0
2666600	Programmer	Category B	CLUSTERPWRDN.PWRDWN is not functional	r0p0, r0p1, r0p2	r1p0
2684598	Programmer	Category B	Software step might go to wrong state	r0p0, r0p1, r0p2	r1p0
2691481	Programmer	Category B	Memory accesses can use old values of some of the Generic System control registers	r0p0, r0p1, r0p2	r1p0
2725097	Programmer	Category B	Interrupting a DSB might cause the DSB for earlier TLBI/IC maintenance operations to fail	r1p0	r1p1
2753598	Programmer	Category B	Deferred error might become uncontainable	r2p0	r3p0
2772412	Programmer	Category B	DFB does not wait for completion of IC/TLBI from other contexts	r1p0	r1p1
2776909	Programmer	Category B	A write to LLRAM following a hardware access flag update could result in incorrect translation	r1p0	r1p1
2795835	Programmer	Category B	A translation table walk may raise an unexpected abort	r1p0	r1p1
3008524	Programmer	Category B	ETM4 not accessible using DENSE_CS_ADDR_MAP configuration	r1p0, r1p1	r2p0

ID	Area	Category	Summary	Found in versions	Fixed in version
3014884	Programmer	Category B	Activation of interrupt might not be delivered to GIC	r0p0, r0p1, r0p2, r1p0, r1p1	r2p0
3100694	Programmer	Category B	Defined interrupt latency bound might be violated	r0p2, r1p0, r1p1	r2p0
3195930	Programmer	Category B	CHI atomic load/cmp/swap with NDERR response and ACP read to same address can lead to deadlock	r2p0, r3p0	r3p1
3299478	Programmer	Category B	DSB SY might deadlock when executed after a context switch	r1p0, r1p1, r2p0	r3p0
3321568	Programmer	Category B	Execution in VMSA might deadlock after entering WFx	r1p0, r1p1, r2p0	r3p0
3394103	Programmer	Category B	Precise ECC error in L1 D-cache tag RAM when using both MM and LLRAM ports might lead to data corruption	r3p0	r3p1
3088388	Programmer	Category B (rare)	Cluster configuration registers can be corrupted after leaving memory retention state	r1p0, r1p1	r2p0
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

ID	Area	Category	Summary	Found in versions	Fixed in version
2224832	Programmer	Category C	Debug APB accesses to reserved core address is accessible when core powered off	r0p0, r1p0, r1p1	r2p0
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
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

ID	Area	Category	Summary	Found in versions	Fixed in version
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	r1p0
2440118	Programmer	Category C	Some PMU events count incorrectly	r0p2	r1p0
2445341	Programmer	Category C	Cache initialization and reset of register fields with UNKNOWN values do not work as expected after Debug recovery	r0p0, r0p1, r0p2	r1p0
2604629	Programmer	Category C	ETM data trace values for ST atomics are incorrect	r0p0, r0p1, r0p2	r1p0
2617173	Programmer	Category C	An external atomic receiving a SLVERR or DECERR might not raise an asynchronous abort	r0p0, r0p1, r0p2	r1p0
2618272	Programmer	Category C	When toggling SPIDEN or DBGEN pins close to a core reset, the system might deadlock	r0p0, r0p1, r0p2	r1p0
2652888	Programmer	Category C	Receiving poison on invalid data lanes might result in a data abort when poison is not supported	r0p0, r0p1, r0p2, r1p0	r1p1
2656238	Programmer	Category C	New interrupts might not be taken after an atomic instruction is interrupted	r0p2	r1p0
2659537	Programmer	Category C	Spurious ECC error might be reported when changing operating mode from DL1ONLY to FULL RAM	r0p0, r0p1, r0p2	r1p0
2661859	Programmer	Category C	Incorrect value reported in RAS registers on LLRAM receiving poison when ECC is not supported	r0p0, r0p1, r0p2	r1p0
2693781	Programmer	Category C	Quality of Service is applied to Main Manager accesses when disabled	r0p1, r0p2	r1p0
2694966	Programmer	Category C	Fault handling interrupt can cause power transition to deadlock	r0p0, r0p1, r0p2	r1p0
2719885	Programmer	Category C	Software step exception might set the ESR_ELx.ISV to a wrong value	r0p0, r0p1, r0p2, r1p0	r1p1
2729228	Programmer	Category C	Debug accesses via APB are not supported in EMU power states	r0p0, r0p1, r0p2, r1p0	r1p1
2729398	Programmer	Category C	L2RDLAT and L2WRLAT fields in IMP_CLUSTERACTLR_EL1 are hardcoded	r0p0, r0p1, r0p2, r1p0	r1p1
2747728	Programmer	Category C	Some cluster PMU events are incorrect	r0p0, r0p1, r0p2, r1p0	r1p1
2753234	Programmer	Category C	An external atomic receiving poison might not raise an asynchronous abort	r0p0, r0p1, r0p2, r1p0	r1p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2763783	Programmer	Category C	Speculative accesses might still be made when LLRAM is disabled	r0p0, r0p1, r0p2, r1p0	r1p1
2773193	Programmer	Category C	32-bit accesses to ERR<n>MISC0 (n= 1-6) will not work as expected	r0p0, r0p1, r0p2, r1p0	r1p1
2780544	Programmer	Category C	Reset of register fields with UNKNOWN values does not work as expected in the presence of Debug recovery mode	r0p0, r0p1, r0p2, r1p0, r1p1	r2p0
2788717	Programmer	Category C	An interrupted atomic operation with bus error to LLRAM may raise unexpected asynchronous abort	r1p0	r1p1
2798198	Programmer	Category C	Incorrect system reset behavior under MEM_RET and MEM_RET_EMU modes for IMP_CLUSTERACTLR_EL1.UBACCLVL	r1p1	r2p0
2842855	Programmer	Category C	AFSR0_ELx might have stale value after taking an SError interrupt	r0p0, r0p1, r0p2, r1p0, r1p1	r2p0
2847045	Programmer	Category C	External read of TCM while CPUHALT<m> is HIGH might see unexpected ECC errors	r1p0, r1p1	r2p0
2849849	Programmer	Category C	An uncontainable error might prevent MM port from making forward progress	r2p0	r3p0
2929491	Programmer	Category C	Dynamic OP_MODE transition can generate two transitions	r0p1, r1p0, r1p1	r2p0
3007301	Programmer	Category C	Auxiliary Fault Status Registers (AFSR0_ELx) can report wrong values corresponding to some Exception Syndrome Registers (ESR_ELx) faults	r1p0, r1p1	r2p0
3025593	Programmer	Category C	Unaligned exclusive to Normal Non-cacheable memory is not aborted	r0p0, r0p1, r0p2, r1p0, r1p1	r2p0
3105480	Programmer	Category C	FAR register should record the lowest address that caused a fault	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0	r3p0
3114083	Programmer	Category C	Cluster OFF_EMU to ON transition stuck because PACCEPT is blocked due to debug requests during OFF_EMU	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0, r3p1	Open
3123474	Programmer	Category C	Some PMU events count incorrectly	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0	r3p0
3196425	Programmer	Category C	Incorrect VA[14:12] sent to Instruction Cache for SnpDVMOp Virtual I-Cache Invalidate from CHI	r2p0	r3p0

ID	Area	Category	Summary	Found in versions	Fixed in version
3197529	Programmer	Category C	AT operation might incorrectly indicate a Synchronous External abort in PAR_EL1	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0	r3p0
3203547	Programmer	Category C	IMP_INTLATENCY_EL2.DEV might cause a Synchronous External abort for IC instructions to Device addresses	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0	r3p0
3338838	Programmer	Category C	IMP_CLUSTERACTLR_EL1.L2SPEC specification is incorrect	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0	r3p1
3378740	Programmer	Category C	ECC error in L1 D-cache tag RAM might not get logged in RAS record	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0	r3p1
3454202	Programmer	Category C	Unused utility bus address locations are aliased	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0	r3p1
3545298	Programmer	Category C	An external Non-cacheable store exclusive receiving a bus error response might not raise an asynchronous abort	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0, r3p1	Open
3606619	Programmer	Category C	L2 Flushes too many ways when moving from FULL_RAM to HALF_RAM	r1p0, r1p1, r2p0, r3p0	r3p1
3639153	Programmer	Category C	Cold debug recovery reset will not reset some system registers	r1p0, r1p1, r2p0, r3p0, r3p1	Open
3683909	Programmer	Category C	The CPU ROM Table reports an incorrect base address for CTI2-7 for configurations with DENSE_CS_ADDR_MAP = 1	r1p0, r1p1, r2p0, r3p0	r3p1
3686657	Programmer	Category C	Interrupt might be taken on a read from device memory that receives a bus error	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0	r3p1
3769700	Programmer	Category C	L2 incorrectly applies address-based hazarding to Non-Reorderable Device writes, reducing performance	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0	r3p1
3776502	Programmer	Category C	L2 incorrectly applies address-based hazarding to Reorderable Device writes, reducing performance	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0, r3p1	Open
3823246	Programmer	Category C	Cluster PMU does not generate overflow	r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0, r3p1	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 might 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, r0p2, r1p0 and r1p1. Fixed in r2p0.

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 might 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, ON->OFF_EMU or ON->MEM_RET
- A Core in the system is requested to transition to a higher power mode, particularly, OFF->ON or OFF_EMU->ON
- The requests 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 (for example, 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 and 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 and 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 and 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 and r0p1. 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 and 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 and 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. Fixed in r1p0.

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. Fixed in r1p0.

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.

2455174
Cluster MEM_RET is not sustained

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

Cortex-R82 supports multiple power modes including the cluster Memory retention power mode (MEM_RET). The system can choose to go into MEM_RET power mode with Dynamic power mode transitions or Static power mode transitions. The device power requirements are indicated by internal Pactive signals.

Due to this erratum, once MEM_RET is entered, the cluster will fail to keep requesting MEM_RET.

Configurations affected

This erratum affects all configurations of Cortex-R82 processor with the L2_RET auxiliary configuration parameter set to 1.

Conditions

This erratum occurs in either of the following cases of power mode transition:

- Through a cluster Static power mode transition and when the following sequence of conditions is met:
  a. CLUSTERPPU_PWPR.PWR_DYN_EN is set to 0
  b. All the cores are powered down (Power OFF mode)
  c. CLUSTERPPU_PWPR.PWR_POLICY is set to 0b0010 (MEM_RET).

- Through a cluster Dynamic power mode transition and when the following sequence of conditions is met:
  a. CLUSTERPPU_PWPR.PWR_DYN_EN is set to 1
  b. CLUSTERPPU_PWPR.PWR_POLICY is set to either of the following:
     - 0b0010 (MEM_RET)
     - 0b0001 (OFF_EMU) or 0b0000 (OFF).
  c. One of the system cores sets IMP_CLUSTERPWRDN_EL1.MEMRET to 1
  d. All the cores are powered down (Power OFF mode)

Implications

When this erratum occurs, the system might indicate an incorrect value in CLUSTERPPU_DISR.PWR_DEVACTIVE_STATUS[3] == 0.

In the conditions of Dynamic power mode transitions when the lowest mode is OFF_EMU or OFF (CLUSTERPPU_PWPR.PWR_POLICY is 0b0001 or 0b0000) the system might unexpectedly further lower its power mode.

Workaround

To avoid this erratum:
If there is a Utility bus connection either via a *System Control Processor* (SCP) or loopback connection:

- For any power mode transition, after entering MEM_RET state, CLUSTERPPU_DISR.PWR_DEVACTIVE_STATUS[3] must not be relied upon to decide to further lower the system power mode.
- If using dynamic power mode transitions to go into MEM_RET state (IMP_CLUSTERPWRDN_EL1.MEMRET == 1), CLUSTERPPU_PWPR.PWR_POLICY should be set to 0b0010 (MEM_RET) prior to setting IMP_CLUSTERPWRDN_EL1.MEMRET to 1. This will avoid the system further lowering its power mode, but CLUSTERPPU_DISR.PWR_DEVACTIVE_STATUS[3] will still indicate an unreliable value while in MEM_RET.

If there is no Utility bus connection:

- MEM_RET should not be used (IMP_CLUSTERPWRDN_EL1.MEMRET should always be 0, default).

2487068
Cluster IRQs may be lost in the presence of cluster MEM_RET and OFF power transitions

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

Cortex-R82 generates *Reliability, Availability, and Serviceability* (RAS) and *Performance Monitoring Unit* (PMU) IRQs at the cluster level. It also supports power modes which put the cluster in OFF or MEM_RET state.

Because of this erratum, IRQs that are generated while handling a power mode transition to OFF or MEM_RET may be lost.

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. One or more of the following is enabled:
    - Cluster PMU interrupts.
    - RAS Critical error interrupt, Fault handling interrupt, or Uncorrected error recovery interrupt.
2. A cluster power mode transition to OFF or MEM_RET is initialized with no cluster interrupt active (that is, nCLUSTERPMUIRQ, nCOMPLEXFAULTIRQ, nCOMPLEXERRIRQ, nCOMPLEXCRITIRQ all set to 1).
3. An event generates an enabled interrupt from step 1.

Implications

When a transition to OFF or MEM_RET completes, any pending cluster interrupt will be de-asserted, and the interrupt information lost.

Workaround

To avoid this erratum, the last powered on core in a Cortex-R82 processor should ensure that the L2 cache is cleaned via software before the core is powered off. This does not prevent missing IRQs, but it is not expected that PMU or RAS IRQs will then contain critical information, because all cores will be OFF and no dirty data will be within the L2 cache.

Alternatively prevent the cluster from going into OFF or MEM_RET power mode by keeping CLUSTERPPU_PWPR default, setting IMP_CLUSTERPWRDN_EL1.MEMRET == 0 (default) and IMP_CLUSTERPWRDN_EL1.PWRDWN == 1

2639772
Turning a core ON while the cluster is being powered OFF may deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor allows power of the shared cluster logic and the cores to be controlled separately. When the cluster logic is being powered OFF from the ON power state, if there is an attempt to power ON one of the cores then it could cause the processor to deadlock.

Configurations affected

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

Conditions

This erratum occurs when all the following conditions are met:

- All cores are powered down
- A power transition from Cluster ON to Cluster OFF is started
- One of the following conditions occur:
  - A dynamic power transition occurs to power a core up due to a COREWAKE interrupt
  - Static programming of the *Power Policy Units* (PPUs) requests one of the cores to power up
- The L2 write back overlaps with the core power transition, 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).

2666600
CLUSTERPWRDN.PWRDWN is not functional

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor controls its power domains through the built-in *Power Policy Unit* (PPU). It also provides software hinting for certain power transitions through CLUSTERPWRDN, for the cases when all cores are in OFF power mode.

Due to this erratum, the PPU is given wrong hints, which can result in an unexpected cluster Power Off when all cores are in an OFF power mode.

Configurations affected

This erratum affects all configurations Cortex-R82.

Conditions

- IMP_CLUSTERPWRDN_EL1.PWRDWN is set to 1 in at least one Cortex-R82 Core, that is, at least one core requests for the cluster to never power itself off.
- All the cores in Cortex-R82 are powered OFF.

Implications

If this erratum occurs, CLUSTERPPU_DISR.PWR_DEVACTIVE_STATUS will display a lower value than expected.

Additionally, if CLUSTERPPU_PWPR.PWR_DYN_EN and CLUSTERPPU_PWPR.PWR_POLICY have default values (i.e., Dynamic Power Mode transitions to OFF are enabled) the cluster will power itself OFF.

Workaround

If the Utility Bus is accessible, CLUSTERPPU_PWPR.PWR_POLICY should be set to 0b1000 (ON power mode) and no hints should be taken from the system through CLUSTERPPU_DISR.PWR_DEVACTIVE_STATUS to decide power mode transitions.

If there is no Utility Bus access, to prevent the cluster from going OFF, there should be at least one core in the system guaranteed to stay ON, i.e., a core should have IMP_CPUPWRCTLR_EL1.PWRDN == 0 (default).

2684598
Software step might go to wrong state

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

Software step is a debug resource that a debugger can use to make the core single-step instructions.

Because of this erratum, the software step might go to a wrong state after executing an exception return instruction or exiting Debug state.

Configurations affected

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

Conditions

This erratum occurs when either of the two following sequences occur:

- Sequence 1

1. Core is executing at EL2 and might be in Debug state.
2. Software step is disabled at EL2 and at target Exception level.
3. The debugger changes HCR_EL2.TGE to 1 to enable software step at target Exception level, but without Context synchronization events.
4. The debugger sets MDSCR_EL1.SS to 1.
5. The debugger executes an Exception return instruction with SPSR_EL2.SS set to 1 or exits Debug state with DSPSR_EL0.SS set to 1.
6. Software step is now in Active-not-pending state.

- Sequence 2

1. Core is executing at EL2 and might be in Debug state.
2. Software step is disabled at EL2 but enabled at target Exception level.
3. The debugger changes HCR_EL2.TGE to 0 to disable software step at target Exception level, but without Context synchronization events.
4. The debugger sets MDSCR_EL1.SS to 1.
5. The debugger executes an Exception return instruction with SPSR_EL2.SS set to 1 or exits Debug state with DSPSR_EL0.SS set to 1.
6. Software step is now in Inactive state.

7. Software step moves to Active-pending state.

Implications

If the first sequence causes this erratum, the PSTATE.SS will not be set to 1 by copying from SPSR_EL2.SS or DSPSR_EL0.SS, so the software step will enter Active-pending state instead of the expected Active-not-pending state. Software Step exception will be generated without stepping the first instruction after returning to core execution.

If the second sequence causes this erratum, the PSTATE.SS will be wrongly set to 1 by copying from SPSR_EL2.SS or DSPSR_EL0.SS. In the last condition, software step will move to Active-not-pending state rather than the expected Active-pending state.

Workaround

The debugger can execute an ISB after programming HCR_EL2.TGE before the Exception return or exiting Debug state.

2691481
Memory accesses can use old values of some of the Generic System control registers

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

Software can set up characteristics of memory accesses through different System control registers. Some of these characteristics are setup using MAIR_EL2, VSTCR_EL2 and VTCR_EL2 registers. When these registers are updated by the software then, after a Context synchronization event, any following memory accesses use the updated values of these registers. This erratum causes the following memory accesses to use the old values of these registers instead of the updated values.

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. Memory accesses use a particular set of values from MAIR_EL2, VSTCR_EL2 and VTCR_EL2 registers
2. Software updates fields of only MAIR_EL2, VSTCR_EL2 or VTCR_EL2 registers that these memory accesses are using
3. Software executes a Context synchronization event (which could include an Exception Return (ERET))
4. New memory accesses use the updated fields of MAIR_EL2, VSTCR_EL2 and VTCR_EL2 registers

Implications

When this erratum occurs, memory accesses use old values of MAIR_EL2, VSCTR_EL2 and VTCR_EL2 registers even after updates to these registers and the Context synchronization event.

Workaround

To avoid this erratum, read and then write back the unmodified value of SCTLR_EL2 register after updating any of the MAIR_EL2, VSCTR_EL2 and VTCR_EL2 registers and before the Context synchronization event that is meant to apply the new values.

2725097
Interrupting a DSB might cause the DSB for earlier TLBI/IC maintenance operations to fail

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

Description:

A DSB guarantees the completion of older *TLB Invalidate* (TLBI) and/or older *Instruction Cache* (IC) maintenance operation before the DSB completes. This erratum causes the DSB to complete before the older TLBI or older IC maintenance operation completes on the *Main Manager* (MM) port.

Configurations Affected:

All configurations are affected.

Conditions:

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

1. TLBI or IC maintenance operation is executed to MM port
2. IC maintenance operation is executed to *Low-latency RAM* (LLRAM) port
    a. Note: (1) and (2) can be in either order for this erratum to occur
3. DSB is executed
4. DSB is interrupted before it completes
5. New TLBI or IC maintenance operation is executed to MM port
6. DSB is executed
7. The above conditions occur along with specific micro-architectural conditions with precise timing

Implications:

Memory access using a TLB entry that has been invalidated by the subsequent TLBI, could not have been completed when the DSB to MM finishes. Instruction cache access that should see the effect of previous IC maintenance operation could not see that affect. Most of the software are not expected to use MM and LLRAM ports in the manner that can cause this issue.

Workaround:

This erratum can be avoided if a DSB is executed after sending the operation (TLBI/IC maintenance to MM or IC maintenance to LLRAM port) to each port, instead of executing the DSB after sending the operation to both ports.

2753598
Deferred error might become uncontainable

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r2p0. Fixed in r3p0.

Description

Poison information cached in a *Processing Element* (PE) might be lost for a memory location on the *Main Manager* (MM) port and therefore make the deferred error to become uncontainable.

Configurations Affected

This erratum affects configurations having all of the following:

1. `RAM_PROTECTION` configuration parameter is set to 1
2. The interconnect does not have a precise snoop filter, and does not use SnpQuery to inquire about the state of the line at the *Request Node* (RN).

Conditions

1. A MM line is cached in the PE and in another cache in the system.
2. A MakeReadUnique from the complex is processed by the interconnect and poisoned data is returned, without the line being lost by the complex.

Implications

If the condition occurs, the line might be propagated to the core without being poisoned.

Workaround

Software can enable the Error Recovery Interrupt for deferred error by setting the DUI bit of all the *Reliability, Availability, and Serviceability* (RAS) node registers ERRxCTLR for node 4 to prevent the error from becoming uncontainable.

2772412
DFB does not wait for completion of IC/TLBI from other contexts

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

Description

The *Data Full Barrier* (DFB) instruction is intended to ensure completion of memory accesses from all contexts. Due to this erratum, DFB may not ensure completion of IC or TLBI operations that were executed in a different context.

Configurations affected

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

Conditions

This erratum occurs if the following sequence of events occurs:

1. A TLBI or IC operation is executed in an Exception level lower than EL2
2. An exception is taken to EL2
3. A DFB is executed in EL2

Implications

If this erratum occurs, the DFB may complete before the preceding TLBI or IC operation. (On returning to the original context a DSB SY will wait for completion of the TLBI or IC, so the original context will not be able to observe old translation tables or old instructions.)

Workaround

This erratum can be avoided by applying the following patch. DFB is not expected to execute often, so any performance impact should be minimal. The code sequence should be applied early in the boot sequence prior to any IC or TLBI ops being executed.

```
mov x0, 0
msr s3_2_c15_c8_0, x0 // IMP_CPUPSELR_EL1
ldr x0, =0xFFFFFFFF
msr s3_2_c15_c8_3, x0 // IMP_CPUPMR_EL1
ldr x0, =0xD5033C9F
msr s3_2_c15_c8_2, x0 // IMP_CPUPOR_EL1
```

```
ldr x0, =0x20100101
msr s3_2_c15_c8_1, x0 // IMP_CPUPCR_EL1
isb
```

2776909
A write to LLRAM following a hardware access flag update could result in incorrect translation

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

Description

When the hardware access flag update is enabled, if there is a context change to a context using *Protected Memory System Architecture* (PMSA) and a write to *Low Latency Random Access Memory* (LLRAM) is executed then it is possible for an incorrect translation result to be cached. Subsequent accesses to the address for which the hardware access flag update was performed could receive an incorrect translation.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 which include *Virtual Memory System Architecture* (VMSA).

Conditions

This erratum occurs if the following sequence of events occurs:

1. An instruction is executed which requires a hardware access flag update
2. An interrupt occurs which is taken to EL2
3. A write to LLRAM is executed in a PMSA context (either EL2 or after context switching to a new EL0/1 PMSA context)
4. Specific micro-architectural conditions occur

Implications

When this erratum occurs, an incorrect translation could be cached in the *Translation Lookaside Buffer* (TLB). This could result in incorrect permissions or attributes used for subsequent accesses to the same address for which the access flag update was performed.

Workaround

To work around this erratum, a DFB instruction should be executed in EL2 after taking an exception from a VMSA context. This DFB instruction should be executed before any writes to LLRAM are performed.

2795835
A translation table walk may raise an unexpected abort

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0. Fixed in r1p1.

Description

Due to this erratum, a translation that is performed for an instruction fetch or data side access might return a translation error that causes an abort to be taken. However, this can only occur when a load exclusive is executed to the same address as the translation table walk.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 which include *Virtual Memory System Architecture* (VMSA).

Conditions

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

1. A load exclusive is executed to the *Main manager* (MM) port:
    - This load is to memory with Normal Inner Write-Back, Outer Write-Back Cacheable attributes
    - This load misses in the L1 cache but the data is held in the L2 cache
    - This might be speculative
2. A translation table walk occurs which is to the same address as the load exclusive was accessing, this could be for:
    - An instruction fetch
    - A different data side memory access if the load exclusive is interrupted
3. Above conditions occur along with specific micro-architectural conditions with precise timing

Implications

When this erratum occurs and the translation table walk is for an instruction fetch, an Instruction Abort will be raised. When this erratum occurs and the translation table walk is for a data side access, that data side access will raise a Data Abort.

Workaround

This erratum can be avoided using Implicit Error Synchronization, by setting SCTLR_EL1.IESB==1, and applying the following patch. The patch code sequence should be applied early in the boot sequence prior to any exclusive operations being executed.

```
mov x0, 1
msr s3_2_c15_c8_0, x0 // IMP_CPUPSELR_EL1
ldr x0, =0xBFE00000
msr s3_2_c15_c8_3, x0 // IMP_CPUPMR_EL1
ldr x0, =0x08400000
msr s3_2_c15_c8_2, x0 // IMP_CPUPOR_EL1
ldr x0, =0x806001C1
msr s3_2_c15_c8_1, x0 // IMP_CPUPCR_EL1
mov x0, 2
msr s3_2_c15_c8_0, x0 // IMP_CPUPSELR_EL1
ldr x0, =0xBFC00000
msr s3_2_c15_c8_3, x0 // IMP_CPUPMR_EL1
ldr x0, =0x88400000
msr s3_2_c15_c8_2, x0 // IMP_CPUPOR_EL1
ldr x0, =0x806001C1
msr s3_2_c15_c8_1, x0 // IMP_CPUPCR_EL1
isb
```

3008524
ETM4 not accessible using DENSE_CS_ADDR_MAP configuration

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0 and r1p1. Fixed in r2p0.

Description

If an access is made to the address range for the *Embedded Trace Macrocell* (ETM) of core4, this will be routed to the ETM for core0.

Configurations affected

This erratum affects configurations of the Cortex-R82 with CS_DENSE_ADDR_MAP set to 1

Conditions

The erratum occurs when a read or write access is made to address 0x20mmm where m is from 0 to F.

Implications

If the processor is implemented with more than 4 cores, the ETM of the 5th core can not be accessed from external debug and instead the ETM for the first core will be accessed. System register access is not affected.

Workaround

There is no workaround. It is possible to confirm the ETM which is accessed by reading the TRCDEVAFF register at offset 0xFA8. Bits [15:8] will report the associated core number.

3014884
Activation of interrupt might not be delivered to GIC

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r1p0 and r1p1. Fixed in r2p0.

Description

The Cortex-R82 processor implements the *Generic Interrupt Controller* (GIC) CPU interface. The interrupt handler reads ICC_IAR0_EL1 or ICC_IAR1_EL1 in the CPU interface to acknowledge the interrupt.

Because of this erratum, the GIC CPU interface might not communicate the activation of interrupt to the external GIC, causing repeated interrupt or higher interrupt latency.

Configurations affected

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

Conditions

This erratum occurs when all the following conditions are met:

- The GIC CPU interface is enabled (**GICCDISABLE** signal LOW at reset)
- Interrupt handler issues MRS ICC_IAR0_EL1/ICC_IAR1_EL1 together with a load/store instruction
- The load/store encounters a stalling condition delaying the read of ICC_IAR0_EL1/ICC_IAR1_EL1 to complete

Implications

If this erratum occurs, the core will not signal the activation of interrupt to the external GIC, and ICC_RPR_EL1, ICC_AP0R_EL1 or ICC_AP1R_EL1 will not be updated for the current interrupt. When the current interrupt is preempted by a higher priority interrupt, repeated interrupt might be seen after returning from processing the higher priority interrupt. In other cases, secondary impact to the interrupt latency of new lower priority interrupts might be seen.

Workaround

To avoid the erratum, the interrupt service routine can place two *No Operations* (NOPs) immediately after the MRS ICC_IAR0_EL1/ICC_IAR1_EL1. This will have neglectable impact on interrupt latency.

A generic handler preamble example included in the Cortex-R82 TRM has the following sequence of code:

```
MRS X0, ICC_IAR0_EL1
STP X4, X5, [SP, #-16]!
MRS X1, SPSR_EL1
STP X6, X7, [SP, #-16]!
```

If the generic handler example has been adopted in your software, this shall be changed to be as follows:

```
MRS X0, ICC_IAR0_EL1
NOP
NOP
STP X4, X5, [SP, #-16]!
MRS X1, SPSR_EL1
STP X6, X7, [SP, #-16]!
```

3100694
Defined interrupt latency bound might be violated

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r0p2, r1p0 and r1p1. Fixed in r2p0.

Description

The Cortex-R82 processor has defined a bounded interrupt latency response under certain conditions.

Because of this erratum, the defined interrupt latency bound might be violated.

Configurations affected

The erratum affects all configurations of the Cortex-R82 processor when used with the Arm *Generic Interrupt Controller* (GIC).

Conditions

This erratum occurs when all the following conditions are met:

- The GIC CPU interface is enabled (GICCDISABLE signal LOW at reset)
- A load instruction is co-issued with a store instruction
- The older load/store instruction of the co-issued pair encounters a stalling condition when targeting the *Low-latency RAM* (LLRAM) region or the *Main Manager* (MM) region
- An interrupt is pending and not masked

Implications

If this erratum occurs, the core will not take the interrupt until the stalled load/store instruction completes. If the stall is caused by a cache miss targeting the MM region, the interrupt latency might be increased significantly depending on the interconnect. If the stall is caused by the memory operation targeting the LLRAM region, the interrupt latency is expected to be slightly increased as the LLRAM port is assumed to be connected directly to an external memory.

Workaround

If interrupt latency is a primary concern, this erratum can be avoided by disabling load/store dual-issue by setting IMP_CPUACTLR_EL1.LSDI to 1.

This will have a performance impact in the order of 3% to general applications and 16% to memory intensive applications. Thus, the workaround shall be used only when the system might be seriously impacted by an increase to the interrupt latency.

3195930
CHI atomic load/cmp/swap with NDERR response and ACP read to same address can lead to deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r2p0 and r3p0. Fixed in r3p1.

Description

The Cortex-R82 processor can optionally send coherent atomic requests externally on the CHI interface when broadcastatomic is set. Due to this erratum, L2 can deadlock from a CPU coherent load/compare/swap atomic never releasing an address hazard.

Configurations affected

This erratum affects configurations of the Cortex-R82 processor which include a main manager CHI interface with broadcastatomics set.

Conditions

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

- A core makes a coherent load, compare or swap atomic that goes out externally on the CHI interface
- The cluster holds a copy of the cache line addressed by the atomic
- The CHI interface returns an NDERR response without performing a snoop based on the TXREQ snoopme bit
- An ACP coherent read request to the same address as the atomic hits in the cluster

Implications

If this erratum occurs then the cluster can maintain an indefinite address hazard on the address used by the CPU coherent atomic. The CPU core will get an NDERR response forwarded to it, but the ACP read request will never complete. A deadlock can occur in the L2 due to the persistent hazarding blocking forward progress of other traffic. The CPU core will take an abort based on the NDERR response it receives.

This is not expected to impact most systems due to getting an NDERR response in combination with timing of an ACP read request to the same location being rare. For interrupt latency it is recommended to force all atomics near which would prevent this erratum from occurring. It is not expected that Cortex-R82 will be used in large systems so forcing atomics near should have minimal performance impact.

Workaround

Force atomics near by setting either IMP_CPUACTLR_EL1.ATOM to 0b01 or broadcastatomic = 0.

3299478
DSB SY might deadlock when executed after a context switch

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0, r1p1 and r2p0. Fixed in r3p0.

Description

Cortex-R82 implements the Arm®v8-R AArch64 architecture which supports context specific barriers, such that DSB SY will only order transactions in the same context. Due to this erratum, the DSB SY might not complete when a context specific DSB SY is executed while certain instructions from a previous context still have associated transactions in progress in the memory system.

Configurations affected

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

Conditions

This erratum occurs if the following sequence of events occurs:

1. An interrupt is taken on the instruction before a Compare and Swap atomic instruction in Context A
2. Complex and unexpected microarchitectural timing conditions occur
3. Context B executes a DSB SY where both of the following are true:
    - Context B also has an outstanding IC or TLBI instruction for which a DSB SY has not previously been executed in Context B prior to this DSB SY. This can be executed before or after the atomic instruction
    - The IC or TLBI instruction is targeting a different port to the interrupted atomic

Implications

If the erratum occurs, the DSB SY will deadlock. The DSB SY will still be interruptible, as long as interrupts are not masked.

In most circumstances, the conditions for this erratum will not be met because the transactions associated with the interrupted atomic will complete before starting to execute the interrupt handler.

Workaround

For r1p0, there is no workaround.

For other revisions, in order to work around this erratum, software running in EL2 should execute a DFB instruction before switching to a new context.

3321568
Execution in VMSA might deadlock after entering WFx

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r1p0, r1p1 and r2p0. Fixed in r3p0.

Description

Cortex-R82 supports use of *Virtual Memory System Architecture* (VMSA) for the EL1 and EL0 translation regimes, and it is configurable if pagetables are cached in the L1 data cache or the L2 cache. Due to this erratum, if the pagetables are configured to be cached in the L2 cache and the processor enters a low-power state due to a WFI or WFE instruction, further pagewalks might not be able to complete.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor configured to include VMSA (Per-core VMSA configuration parameter is set to 1).

Conditions

This erratum occurs if the following sequence of events occurs:

1. Execution is in VMSA mode and IMP_CPUACTLR_EL1.PTCL == 1
2. A pagewalk is started due to a TLB miss
3. A WFI or WFE instruction is executed
4. A low-power state is entered before the pagewalk completes

Implications

If this erratum occurs, further pagewalks might not be able to make forward progress and so further execution in VMSA might not be possible. Execution at EL2 or in a *Protected Memory System Architecture* (PMSA) context is unaffected when this erratum occurs.

Workaround

In order to work around this erratum, set IMP_CPUACTLR_EL1.PTCL == 0.

3394103
Precise ECC error in L1 D-cache tag RAM when using both MM and LLRAM ports might lead to data corruption

Status

Affects: Cortex-R82
Fault Type: Programmer Category B
Fault Status: Present in r3p0. Fixed in r3p1.

Description

The L1 data cache tag RAM provides SEDDED protection against errors, because of this erratum when an error occurs in specific bits of the L1 data cache tag RAM it might results in data corruption from some cache states.

Configurations affected

This erratum affects configurations of the Cortex-R82 processor where it is configured to include RAM protection (**`RAM_PROTECTION`** configuration parameter is set to 1) and the *Low-latency RAM* (LLRAM) port is implemented (**`LLRAM`** configuration parameter is set to 1).

Conditions

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

1. Cachelines for the *Main Manager* (MM) port are stored in the L1 data cache
2. A cache lookup for a load instruction occurs for an LLRAM address which is not in the cache
3. An error occurs in the L1 data cache tag RAM which appears to change the state of a MM cacheline from valid to invalid
4. The linefill for the LLRAM lookup does not allocate into the cache
5. Another cache lookup is performed for the same LLRAM cacheline
6. Additional micro-architectural conditions occur

Implications

When this erratum occurs, the linefill for the second cache lookup overwrites a valid cacheline for the MM port. This might cause silent data corruption. The error is not reported to RAS, but would be reported if a MM cache lookup was performed for the same set.

Workaround

In order to work around this erratum, set IMP_CPUACTLR_EL1.DCWT (bit 36) to 0. This might have a small power impact on 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.DCWT=1 for sample silicon.

Category B (rare)

3088388
Cluster configuration registers can be corrupted after leaving memory retention state

Status

Affects: Cortex-R82
Fault type: Programmer Category B (rare)
Fault status: Present in r1p0 and r1p1. Fixed in r2p0.

Description

The Cortex-R82 processor provides a mechanism for maintaining the values of several cluster configuration registers as the cluster enters and leaves the MEM_RET (or MEM_RET_EMU) state. As a result of this erratum, it is possible for the values which are restored to be incorrect.

Configurations affected

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

Conditions

The erratum occurs when writes to the following registers are made with different values in close succession

- IMP_CLUSTERPWRCTLR_EL1.PRTNRQ, IMP_CLUSTERPWRDN_EL1.MEMRET, IMP_CLUSTERACTLR_EL1.UBACCLVL
  Followed by entry to MEM_RET or MEM_RET_EMU, and then entry to ON state.

Implications

When this erratum occurs, the value of the affected registers after leaving the memory retention state is unpredictable.

Workaround

The erratum can be avoided by repeating the access if different values must be written to any of these registers, the final value can be written again unchanged.

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: Present in 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, r1p0 and r1p1. Fixed in r2p0.

Description

If a read or write transaction is made on the Debug *Advanced Peripheral Bus* (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

For release r0p0, this erratum affects all configurations of the Cortex-R82.
For releases r1p0 and r1p1, this erratum affects configurations of Corex.R82 with CS_DENSE_ADDR_MAP set to 1.

Conditions

For release r0p0, 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.
For other affected releases, the erratum occurs when a read or write access is made on the Debug APB interface that is to an address 0xnm_mD90 where n is from 0 to 3 and m is from 0 to F.

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 and 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 and 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 and 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 and 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 and 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 and 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 and 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 and 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. Fixed in r1p0.

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.

2440118
Some PMU events count incorrectly

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p2. Fixed in r1p0.

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:

- 0x0019 BUS_ACCESS
- 0x0060 BUS_ACCESS_RD
- 0x0061 BUS_ACCESS_WR
- 0x021D SPP_CYCLES

Implications

If this erratum occurs:

- Any count of BUS_ACCESS, BUS_ACCESS_RD or BUS_ACCESS_WR would be 0. Software will not be able to use these events for performance analysis.
- Any count of SPP_CYCLES would be incorrect. Software will not be able to use these events for performance analysis.

Workaround

There is no workaround.

2445341
Cache initialization and reset of register fields with UNKNOWN values do not work as expected after Debug recovery

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

Cortex-R82 contains hardware that initializes programmer-visible state when a reset is applied, unless the reset is because of Debug recovery. Specifically, caches are invalidated, and the register file and certain System registers are initialized to fixed values, despite the fact they are specified to reset to UNKNOWN values.

Because of this erratum, this feature may not work as expected.

Configurations affected

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

Conditions

This erratum occurs on any reset caused by Debug recovery.

Implications

With regard to register initialization:

- On Debug recovery with power-on reset, all registers inside the CPU power domain are initialized and therefore do not retain their values as expected.
- On Debug recovery with either power-on or warm reset, the following register families are initialized and therefore do not retain their values as expected:
  - TCR_EL1, TCR_EL2, VSCTLR_EL2, VSTCR_EL2, VTCR_EL2, MAIR_EL1, MAIR_EL2, CONTEXTIDR_EL1, CONTEXTIDR_EL2, PAR_EL1, PMCID1SR, PMCID1SRa, PMCID2SR, PMVIDSR, CPACR_EL1, CPTR_EL2, HCR_EL2, IMP_BPCTLR_EL1, SCTLR_EL1, SCTLR_EL2
  - Core and Cluster Performance Monitor (PMU) registers
  - *Reliability, Availability, and Serviceability* (RAS) registers (on power-on reset only)
  - Special purpose register SPSR_EL2
  - *Generic Interrupt Controller* (GIC) CPU interface system registers

With regard to cache invalidation:

- On Debug recovery with power-on reset, the L2 cache and L1 Duplicated tag RAMs is invalidated.

Workaround

There is no workaround for the affected registers for register initialization.

The L2 cache invalidation works as expected if Debug recovery is used with warm reset. Leave CLUSTERPPU_PTCR.DBG_RECOV_PORST_EN to 0, which is the default value. For correct operation, ensure the value of the core's PPU_PTCR.DBG_RECOV_PORST_EN matches the value set for the *Cluster Power Policy Unit* (PPU).

2604629
ETM data trace values for ST atomics are incorrect

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor implements the *Embedded Trace Macrocell* (ETM). The ETM generates trace for instruction and data transfers.

Because of this erratum, the ETM the data trace generated for the ST versions of atomic instructions (for example STADD) is incorrect.

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 data trace is active
- The core executes an atomic instruction with WZR or XZR as target

Implications

If this erratum occurs, the ETM will generate incorrect data trace P1/P2 pairs with index 0 and 1.

Index 0 data is not expected to be traced (the data read from memory is not available to trace) but will contain the operand data.

Index 1 data is expected to contain the operand data but will contain an UNKNOWN value.

No trace is generated for index 2, this is as expected.

The atomic instructions which load data from memory into a register (for example LDADD) are not affected by this erratum.

Workaround

There is no workaround.

2617173
An external atomic receiving a SLVERR or DECERR might not raise an asynchronous abort

Status

Affects: Cortex-R82

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor supports sending Non-cacheable or Device atomics on the *Main Manager* (MM) AXI port. If one of these atomics gets an error response, the atomic might only abort synchronously when it cannot be guaranteed that memory has not been updated.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor where the system supports atomics on the MM AXI port (**`BROADCASTATOMICM`** == 1).

Conditions

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

1. An atomic instruction that returns data is executed with Non-cacheable or Device attributes
2. The atomic is sent on the MM AXI port
3. An error occurs resulting in a RRESP for the atomic of SLVERR or DECERR
4. The BRESP for the atomic is OK
5. The memory targeted by the atomic is still modified despite the previous error occurring

Implications

If this erratum occurs a synchronous exception will be raised despite the associated memory location having been updated.

Workaround

No workaround is expected to be needed as it is expected that for most systems the response would be the same for both BRESP and RRESP.

2618272
When toggling SPIDEN or DBGEN pins close to a core reset, the system might deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor implements different core power modes which reset the CPU, with OFF power mode being a mode susceptible to be used for power savings.

Cortex-R82 further includes pins to enable invasive debug (DBGEN) and secure privilege invasive debug (SPIDEN).

Because of this erratum, toggling these pins in the proximity of a power transition may result in a deadlock.

Configurations affected

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

Conditions

This erratum occurs when all the following conditions are met:

- DBGEN or SPIDEN pins toggle
- A core in Cortex-R82 executes a power transition to OFF, either dynamically or statically programmed
- Complex micro architectural timing events happen

Implications

If this erratum occurs, the system might become unresponsive.

Workaround

There is no workaround.

2652888
Receiving poison on invalid data lanes might result in a data abort when poison is not supported

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

The Cortex-R82 processor supports poison on the *Main Manager* (MM) AXI interface in order to defer errors. When poison is indicated on an invalid data lane for a read and the associated read data is consumed by an agent that does not support poison it might result in a spurious data abort. This can be for a request from a processor when *Error Correcting Code* (ECC) protection is not enabled, or a request on the *Main Accelerator Coherency Port* (MACP) which does not support poison.

Configurations affected

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

Conditions

This erratum occurs when all the following conditions are met:

- A read request on the MM interface has **RPOISONM** asserted for an invalid data lane
  - Poison is at a granularity of 64-bits, so an invalid data lane is a 64-bit chunk where no bytes are accessed by the transaction
- At least one of the following conditions is met:
  - The processor is configured without RAM protection (`RAM_PROTECTION=0`)
  - RAM protection is disabled (`IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN=0`)
  - The data is for a MACP request

Implications

When this erratum occurs due to cache protection being disabled an exception will be raised when the associated data is consumed by a load. This is not expected to have significant implications as it is not an expected use case to have a system configured with poison support while the processor does not have any RAM protection configured or enabled.

When this erratum occurs for a MACP request, the transaction will return a SLVERR response. This is not expected to have significant implications as the MACP port is optimized for cacheline transactions and so it is not expected that transactions will have invalid data lanes.

Workaround

No workaround is required for this erratum.

2656238
New interrupts might not be taken after an atomic instruction is interrupted

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor implements features to improve the interrupt latency. Atomic instructions once started can be interrupted if certain conditions are met.

Because of this erratum, after an atomic instruction is interrupted and a non-typical interrupt handler which contains no memory access operations is used, new interrupts might not be taken.

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 executes an atomic instruction
2. An interrupt is taken and the atomic instruction is interrupted
3. An interrupt handler which contains no load or store instruction is used

Implications

If this erratum occurs, any new interrupts will not be taken until a load or store instruction is executed.

Workaround

No workaround is expected to be required as the interrupt handler normally contains memory access operations which will unlock the core from taking new interrupts.

2659537
Spurious ECC error might be reported when changing operating mode from DL1ONLY to FULL RAM

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

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. Due to this erratum, under specific conditions it is possible for an *Error Correcting Code* (ECC) error to be spuriously reported in the RAS registers when the L2 cache is powered up by a transition to the FULL RAM operating mode.

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. The L2 cache is powered up (in the FULL RAM operating mode)
2. The L2 cache contains only 1 or 2 valid cachelines
    - One of these cachelines is at index 0 in the cache
3. The L2 cache is powered down (changed to the DL1ONLY operating mode)
4. The L2 cache is powered up again (changed to the FULL RAM operating mode)

Implications

If this erratum occurs, a single or double bit ECC error might be reported for index 0 of the L2 cache Tag RAM in the error record register 4, 5 or 6.
There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

Workaround

No workaround is required for this erratum as it is expected that when the L2 cache is enabled and in use it will contain more than 2 valid cachelines.

2661859
Incorrect value reported in RAS registers on LLRAM receiving poison when ECC is not supported

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor supports receiving poisoned data on the *Low-latency RAM* (LLRAM) port and will report this to the *Reliability*, *Availability*, *and Serviceability* (RAS) registers when *Error Correcting Code* (ECC) is not supported by the core and so the error cannot be deferred. Because of this erratum, the value reported in ERR0MISC0.MEM will be incorrect when this error is reported.

Configurations affected

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

Conditions

This erratum occurs when RAM protection is disabled (IMP_CLUSTERMEMPROTCTLR_EL1.MEMPROTEN=0) or not present (RAM_PROTECTION configuration parameter is set to 0) and the following condition occurs:

- Poisoned data is received on the LLRAM AXI port

Implications

If this erratum occurs, instead of being set to 0x48 as expected, the information in the ERR0MISC0.MEM fields will be a Reserved value rather than any of the valid values for this field. Bit[4] will always be 1 which does not occur for any valid value of ERR0MISC0.MEM, which are 0x40, 0x41, 0x43, 0x44 and 0x48.

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 errors that cannot be deferred in the error record registers.

Workaround

No workaround is required for this erratum.

2693781
Quality of Service is applied to Main Manager accesses when disabled

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor contains a mechanism to provide increased Quality of Service (QoS) to a particular core in the processor. Due to this erratum, when the QoS feature is not enabled it will still be applied to the core configured by `IMP_CLUSTERQOSR_EL1.COREQOSID`.

Configurations affected

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

Conditions

This erratum occurs when all the following conditions are met:

- The QoS feature is not enabled, that is `IMP_CLUSTERQOSR_EL1.COREQOSEN=0`
- The value in `IMP_CLUSTERQOSR_EL1.COREQOSID` corresponds to a valid core in the cluster

Implications

When this erratum occurs, QoS is applied despite not being enabled. If `IMP_CLUSTERQOSR_EL1` has not been written then QoS is applied to core 0.

Workaround

No workaround is required for this erratum.

2694966
Fault handling interrupt can cause power transition to deadlock

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r1p0.

Description

The Cortex-R82 processor implements the *Reliability*, *Availability*, *and Serviceability* (RAS) extension, with RAS System Architecture 1.1. The processor can be configured to raise a Fault Handling Interrupt request through an external pin in the presence of certain types of RAS errors.

Because of this erratum, Fault Handling Interrupts might not work as expected. When the software is attempting to clear errors from a record simultaneously with a new error arriving, it might be possible that the ERR<n>STATUS is cleared but the Fault Handling Interrupt pin is set high for that node. The Fault Handling Interrupt pin might be unable to be cleared by software unless a new error occurs. Additionally, if errors occur while Fault Handling Interrupts are disabled, enabling them might not cause an interrupt to be triggered for records that are already recorded.

Configurations affected

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

Conditions

This erratum occurs when all the following conditions are met:

- Fault Handling Interrupts are enabled for a RAS node, with ERR<n>CTLR.FI and/or ERR<n>CTLR.CFI set to 1, where <n> is 0, 1, or 4.
- The RAS node is fully populated by errors:
  - For RAS record 0 to be affected, the single record must be populated.
  - For RAS records 1-3 and 4-6 (available when RAM_PROTECTION=1) to be affected, all three records must be populated with errors, with at least two of the three records with errors which do not cause a Fault Handling Interrupt (for example, corrected errors but corrected counter not overflown).
- The RAS software reads a RAS record and attempts to clear an error by writing the appropriate value into ERR<n>STATUS
- At the same time as the error is being cleared, a new error that can cause a Fault Handling interrupt occurs and is recorded into the RAS record being cleared.

Implications

When this erratum occurs, the Fault Handling interrupt pin for a core or for the cluster might be asserted, although the corresponding ERR<n>STATUS will be clear. The interrupt will be unable to be cleared by software unless a new error of the same type occurs.

A side effect is that while a Fault Handling Interrupt is active the affected core or cluster will deny requests to power down.

Workaround

The fault interrupt handler should be modified as follows:

- Disable Fault Handling interrupt for RAS errors by clearing the ERR<n>CTLR.FI and ERR<n>CTLR.CFI bits
- Execute an ISB instruction to ensure the write has completed
- Clear the ERR<n>STATUS normally
- Re-enable Fault Handling interrupts by restoring the ERR<n>CTLR.FI and ERR<n>CTLR.CFI bits to their original values
- Re-check ERR<n>STATUS for any new errors that occurred in this window, since new errors recorded while Fault Handling interrupts are disabled might not cause the Fault Handling Interrupt to be set when re-enabled.

2719885
Software step exception might set the ESR_ELx.ISV to a wrong value

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

Software Step is a debug resource that a debugger can use to make the core single-step instructions.

Because of this erratum, the Software Step exception might set the ESR_ELx.ISV to a wrong value after Software Step enters the active-pending state from the inactive state.

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 sets MDSCR_EL1.SS to 1 and executes an exception return instruction setting PSTATE.SS to 1. Software Step is now in Active-not-pending state.
2. The core takes an exception to an Exception level that debug exceptions are disabled from. Software Step is now in Inactive state.
3. The core executes instructions changing the debug target Exception level to be the same as the current Exception level and setting the PSTATE.D to 0. Software Step is now in Active-pending state.
4. Software takes the Software Step exception.

Implications

If this erratum occurs, the ESR_ELx.ISV will be set to 1 and ESR_ELx.EX will be set to 0 though architecture requires ESR_ELx.ISV to be set to 0 when Software Step enters the active-pending state directly from the inactive state.

Workaround

There is no workaround.

2729228
Debug accesses via APB are not supported in EMU power states

Affects: Cortex-R82

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

Cortex-R82 includes *Advanced Peripheral Bus* (APB) interface for debug. Cortex-R82 also features an internal *Power Policy Unit* (PPU) which allows setting the system, and in particular the cluster, to *Off Emulated* (OFF_EMU) and *Emulated Memory Retention Mode* (MEM_RET_EMU) power modes. Under these emulated modes, it should still be possible to access debug components in the system.

Due to this erratum, in these emulated modes, debug accesses under APB are not possible.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when the cluster is configured to a power policy of OFF_EMU or MEM_RET_EMU and an APB access is made to any cluster or core component, for example, cluster and core ROM tables, though this applies to any component.

Implications

When this erratum occurs a SLVERR is returned on the APB bus.

Workaround

There is no workaround for this erratum.

2729398
L2RDLAT and L2WRLAT fields in IMP_CLUSTERACTLR_EL1 are hardcoded

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

IMP_CLUSTERACTLR_EL1 is an **IMPLEMENTATION DEFINED** register which contains control bits that affect the cluster behavior. Two of the fields in this register L2RDLAT and L2WRLAT were provided so that software can control the L2 cache data RAM latencies. These fields are reset to the values set by the global configuration parameters L2_DATA_RD_LATENCY and L2_DATA_WR_LATENCY respectively. The software would be able to write to these fields and read the corresponding values before L2 cache begins operating. Additionally, the write values would change the L2 cache data RAM latencies. Because of this erratum, writes to these fields are ignored and that they have no effect.

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 occurs:

1. Cortex-R82 cluster is brought out of reset and L2 cache is not yet being used.
2. An MSR instruction to IMP_CLUSTERACTLR_EL1 register with intention to change the values of L2RDLAT and L2WRLAT.

Implications

- Use cases which rely on this feature will not work.
- An MRS instruction to IMP_CLUSTERACTLR_EL1 will return the values of global configuration parameters L2_DATA_RD_LATENCY and L2_DATA_WR_LATENCY for the fields L2RDLAT and L2WRLAT respectively
- Hardware would use the L2_DATA_RD_LATENCY and L2_DATA_WR_LATENCY values to control the L2 cache data RAM latencies.

Workaround

No workaround is needed for this erratum, as it is expected that the values of the L2_DATA_RD_LATENCY and L2_DATA_WR_LATENCY should be set to safe values that work for the implemented L2 cache data RAMs.

2747728
Some cluster PMU events are incorrect

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

MM_PREFETCH_CPU_MATCH (0x472), MM_PREFETCH_CPU_KILL (0x473), MACP_STASH_ACP_MATCH (0x165), MACP_STASH_ACP_KILL (0x166), LLRAM_ACCESS_SHARED (0x762), LLRAM_ACCESS_NOT_SHARED (0x763) are **IMPLEMENTATION DEFINED** *Performance Monitoring Unit* (PMU) events which are used to monitor certain behavior of the stash requests. Due to the erratum, the *MATCH events will never be triggered, and the *KILL events will be triggered but not correspond to the stash actually being dropped which means that the event is incorrect. For the LLRAM_ACCESS events, all *ACE-Lite Subordinate* (ACELS) accesses are counted as shared so LLRAM_ACCESS_NOT_SHARED will not count, and LLRAM_ACCESS_SHARED will count non-shared ACELS accesses.

Configurations affected

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

Conditions

This erratum occurs if one or more of the following scenarios occur:

Scenario 1 - The following sequence of conditions is met:

1. PMU counters are selected to count MACP_STASH_ACP_MATCH/MM_PREFETCH_CPU_MATCH
2. A read request is issued by one of the CPUs to an address A
3. A stash request is sent from one of the CPUs or through ACP port to the same address A

Scenario 2 - The following sequence of conditions is met:

1. PMU counters are selected to count MACP_STASH_ACP_KILL/MM_PREFETCH_CPU_KILL
2. There is no L2 cache, L2 cache is powering down, or L2 cache is powered down
3. A stash request is sent from one of the CPUs or through ACP port

Scenario 3 - The following sequence of conditions is met:

1. PMU counters are selected to count LLRAM_ACCESS_SHARED/LLRAM_ACCESS_NOT_SHARED
2. An LLRAM Non-shareable access occurs via the ACELS port

Implications

If this erratum occurs:

- Use cases which rely on counting these PMU events will not be able to do so
- MM_PREFETCH_CPU_MATCH, MACP_STASH_ACP_MATCH will never count
- MM_PREFETCH_CPU_KILL, MACP_STASH_ACP_KILL will count incorrectly
- Any ACELS accesses will be counted in LLRAM_ACCESS_SHARED, regardless of Shareability

Workaround

No workaround is required.

2753234
An external atomic receiving poison might not raise an asynchronous abort

Status

Affects: Cortex-R82

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

The Cortex-R82 processor supports sending Non-cacheable or device atomics on the *Main Manager* (MM) AXI port. If one of these atomics gets poison on the load data, the atomic might only abort synchronously when it cannot be guaranteed that memory has not been updated.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor where the system supports atomics on the MM AXI port (**`BROADCASTATOMICM`** == 1).

Conditions

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

1. An atomic instruction that returns data is executed with Non-cacheable or device attributes
2. The atomic is sent on the MM AXI port
3. An error is detected after memory is updated resulting in poison on the load data
4. The BRESP for the atomic is OK

Implications

If this erratum occurs a synchronous exception will be raised despite the associated memory location having been updated. Moreover atomics to Non-cacheable and device memory is not architecturally guaranteed, so only limited cases will make use of this.

Workaround

No workaround is expected to be needed.

2763783
Speculative accesses might still be made when LLRAM is disabled

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description

The *Low-Latency RAM* (LLRAM) port provides controls to allow enabling and disabling transactions to be made. Due to this erratum, it is possible for speculative accesses to be made to the LLRAM port when it is disabled.

Configurations Affected

This erratum affects all configurations of the Cortex-R82 which implement the LLRAM port.

Conditions

This erratum occurs if the following sequence of events occurs:

1. An MPU region or page is configured corresponding to the physical address of LLRAM
    - This must have read permissions granted
2. The LLRAM region is disabled, IMP_LLRAMREGIONR_EL1.ENABLE = 0
3. A load is executed speculatively to the same physical address as the page previously configured
4. Specific micro-architectural conditions occur

Implications

A speculative access will be performed to the LLRAM when this erratum occurs. It is expected that when the LLRAM port is present and translations are set up to give access to it, that a speculative access to the port will not cause any functional issues.

Workaround

No workaround is required for this erratum.

2773193
32-bit accesses to ERR<n>MISC0 (n= 1-6) will not work as expected

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2 and r1p0. Fixed in r1p1.

Description:

Cortex-R82 implements *Reliability*, *Availability*, *and Serviceability* (RAS) records. It also implements a 64-bit wide Utility bus that allows an external agent to access internal RAS registers.

Due to this erratum, 32-bit accesses to the 64-bit ERR<n>MISC0 (n= 1-6) via the Utility bus might not work as expected.

Configurations affected:

This erratum affects configurations of Cortex-R82 with RAM_PROTECTION = 1.

Conditions:

This erratum occurs when 32-bit accesses are made to the 64-bit register ERR<n>MISC0 (n= 1-6) via the Utility bus.

Implications:

- Reads to the lower half work correctly
- Reads to the upper half return 0
- Writes to the lower half work correctly for written data, but will also zero out the upper half
- Writes to the upper half have no effect

Workaround:

A workaround is not expected to be needed for most software applications. Impacted reads and writes to the upper half of these registers contain the correctable error count (CEC), error type, and overflow (OF), but the ERR<n>STATUS.CE bit will still show if at least one correctable error exists as well as overflow information.

If a workaround is needed, an external agent can access these registers by performing a 64-bit read.

Software running on the core is not affected by this erratum. That is, access to all ERR<n>MISC0 registers through ERXMISC0_EL1 works as expected.

2780544
Reset of register fields with UNKNOWN values does not work as expected in the presence of Debug recovery mode

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0 and r1p1. Fixed in r2p0.

Description

Cortex-R82 contains hardware that initializes programmer-visible state when a reset is applied, unless the reset is because of Debug recovery mode. Specifically, the register file and certain System registers are initialized to fixed values, despite the fact they are specified to reset to UNKNOWN values, except if the reset is due to Debug recovery mode in which case they retain the previous value.

Because of this erratum, this feature might not work as expected.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor in which the processor application relies on persistence of values of System registers with UNKNOWN reset values under debug recovery.

Conditions

This erratum occurs on any Debug recovery mode exit for the cores or cluster.

Implications

On any reset of the listed registers' fields (and, if applicable, their external views) which reset to UNKNOWN might not retain their values when exiting Debug recovery mode:

- PMEVCNTRn
- ICH_VMCR_EL2
- ICV_PMR_EL1
- PMOVSSET_EL0
- PMOVSCLR_EL0
- CNTKCTL_EL1
- CNTP_CTL_EL0
- CNTV_CTL_EL0
- CNTHCTL_EL2
- CNTHPS_CTL_EL2
- CNTVOFF_EL2

- CNTP_CVAL_EL0
- CNTV_CVAL_EL0
- CNTHPS_CVAL_EL2
- IMP_CLUSTERPMSELR_EL1
- IMP_CLUSTERPMCCNTR_EL1
- IMP_CLUSTERPMCNTENCLR_EL1
- IMP_CLUSTERPMCNTENSET_EL1
- IMP_CLUSTERPMINTENCLR_EL1
- IMP_CLUSTERPMINTENSET_EL1
- IMP_CLUSTERPMOVSCLR_EL1
- IMP_CLUSTERPMOVSSET_EL1
- IMP_CLUSTERPMXEVTYPER_EL1
- IMP_CLUSTERPMXEVCNTR_EL1
- ERR<n>CTLR

Workaround

There is no workaround.

2788717
An interrupted atomic operation with bus error to LLRAM may raise unexpected asynchronous abort

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1.

Description:

A bus error on *Low-latency RAM* (LLRAM) port during an atomic operation is expected to generate an asynchronous abort. However, if the atomic operation is interrupted any following bus error for that atomic operation is not expected to generate asynchronous abort. Due to this erratum, an asynchronous abort is generated due to the bus error on LLRAM port for the atomic operation that is already interrupted.

Configurations Affected:

All configurations with *Low-latency RAM* (LLRAM) implemented.

Conditions:

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

1. An atomic instruction is executed to LLRAM
2. Atomic operation is interrupted before it completes
3. BRESP, other than OK, is received for the atomic operation
4. Above conditions occur along with specific micro-architectural conditions with precise timing

Implications:

Due to this erratum, an asynchronous abort is raised. The interrupt and the bus error that generate the asynchronous abort are independent asynchronous events. Software cannot tell whether the bus error happened before or after the interrupt and hence whether the asynchronous abort was expected or not.

Workaround:

No workaround is expected to be required for this erratum.

2798198
Incorrect system reset behavior under MEM_RET and MEM_RET_EMU modes for IMP_CLUSTERACTLR_EL1.UBACCLVL

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r1p1. Fixed in r2p0.

Description

Cortex-R82 supports multiple power modes including the cluster Memory retention power mode (MEM_RET) and an emulated counterpart (MEM_RET_EMU). There exists a configuration field in a System register to configure Utility Bus privilege accesses (IMP_CLUSTERACTLR_EL1.UBACCLVL).

Due to this erratum, once MEM_RET or MEM_RET_EMU is entered, the behavior of the system might mismatch the configuration of this field.

Configurations affected

This erratum affects all configurations of Cortex-R82 processor.

Conditions

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

1. IMP_CLUSTERACTLR_EL1.UBACCLVL has a value different than its reset value
2. The system enters a Memory Retention state (MEM_RET or MEM_RET_EMU)

Implications

The Utility Bus access privileges will not be reset and will retain their previously programmed more restrictive value. Software reading the IMP_CLUSTERACTLR_EL1.UBACCLVL field will see the reset value instead.

Workaround

To avoid this erratum:

Memory retention states will not reset the behavior of IMP_CLUSTERACTLR_EL1.UBACCLVL. The software should not rely on the first read of this register after reset and should instead only rely on its value after a write.

2842855
AFSR0_ELx might have stale value after taking an SError interrupt

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0 and r1p1. Fixed in r2p0.

Description

The Cortex-R82 processor implements AFSR0_ELx registers to record additional IMPLEMENTATION DEFINED fault status information when taking exceptions.

Because of this erratum, if a new SError interrupt was signaled at the same time as the previous SError interrupt was taken, then AFSR0_ELx might contain stale value when the new SError interrupt is taken.

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. An SError interrupt is pended because of an error in LLRAM, SPP or MM
2. The SError interrupt is taken and marks ESR_ELx.AET as 'Uncontainable'; at the same time, a new SError interrupt is pended because of a different type of error in LLRAM, SPP or MM
3. The new SError interrupt is taken

Implications

If this erratum occurs, the AFSR0_ELx associated with the new SError interrupt will have the stale value of the previous SError interrupt.

Workaround

There is no workaround.

2847045
External read of TCM while CPUHALT<m> is HIGH might see unexpected ECC errors

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r1p0 and r1p1. Fixed in r2p0.

Description

The Cortex-R82 processor implements IMP_MEMPROTCTLR_EL1.MEMPROTEN to control RAM protection functionality. The reset value of this control takes the value of the top-level pin CFGRAMPROTEN.

Because of this erratum, if there is an external read of *Tightly Coupled Memory* (TCM) during TCM preload, the RAM protection control reset value is ignored and unexpected *Error Correcting Code* (ECC) errors might be received.

Configurations affected

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

Conditions

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

1. CFGRAMPROTEN is set to 0
2. Out of reset, CPUHALT<m> is held HIGH for TCM preloading
3. There is an explicit read through ACELS to TCM
4. ECC errors are detected

Implications

When this erratum occurs, and with RAM protection disabled:

- The external read will see the corrected value when there is a correctable ECC error
- The external read will see a SLVERR response when there is a non-correctable ECC error

Note as ERR<n>CTLR.ED is reset to 0, there will be no errors recorded to RAS registers.

Workaround

No workaround is expected. The typical case of using TCMs from reset is through write-only *Direct Memory Access* (DMA) to TCMs and ECC errors will be suppressed.

2849849
An uncontainable error might prevent MM port from making forward progress

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r2p0. Fixed in r3p0.

Description

An uncontainable error detected when a core is doing a line upgrade to the *Main Manager* (MM) port might deadlock any further accesses to the MM port.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A core is doing a store to the MM port that requires a line upgrade.
2. Unlikely, timing-sensitive, microarchitectural conditions occur, including an uncontainable error detected in the L2 duplicate L1 tag RAMs.

Implications

There is still substantial benefit being gained from the *Error Correcting Code* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions occur, accesses to the MM port might not be able to make forward progress.

Workaround

There is no workaround.

2929491
Dynamic OP_MODE transition can generate two transitions

Status

Affects: Cortex-R82
Fault type: Programmer Category C
Fault status: Present in r0p1, r1p0 and r1p1. Fixed in r2p0.

Description

The Cortex-R82 processor provides a mechanism for dynamic control of device operating mode via the cluster *Power Policy Unit* (PPU). As a result of this erratum, the first transition from cluster OFF or MEM_RET can incorrectly be split into two power transition sequences, increasing the duration of this transition.

Configurations affected

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

Conditions

The erratum occurs when all the following register values are set:

1. CLUSTERPPU_PWPR.OP_DYN_EN = 0b1
2. CLUSTERPPU_PWPR.OP_POLICY = NO_CACHE

Implications

Each transition from a cluster reset state (Cluster OFF or MEM_RET) will first transition to the L2 cache disabled state, then transition to the L2 cache enabled state (the default value requested by IMP_CLUSTERPWRCTRL_EL1).

Workaround

No workaround is necessary.

3007301
Auxiliary Fault Status Registers (AFSR0_ELx) can report wrong values corresponding to some Exception Syndrome Registers (ESR_ELx) faults

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r1p0 and r1p1. Fixed in r2p0.

Description:

Cortex-R82 has *Exception Syndrome Registers* (ESR_ELx) for syndrome information and it has *Auxiliary Fault Status Registers* (AFSR0_ELx) which provide corresponding additional implementation defined fault status information, where required. Not all the values of ESR_ELx require additional information and for these, corresponding AFSR_ELx register has no useful information. In these cases AFRS0_ELx.PORT reports "Unknown" and AFSR0_ELx.TYPE reports "Other Error".

With this erratum, when ESR_ELx reports Address size, Translation or Permission fault, although no corresponding useful information is required from AFSR0_ELx, it can wrongly set AFSR0_ELx.PORT to "Overlapping illegal port", instead of "Unknown" and it can set AFSR0_ELx.TYPE to "Error on data", instead of "Other Error".

Configurations Affected:

This erratum affects all configurations of the Cortex-R82 which include *Virtual Memory System Architecture* (VMSA).

Conditions:

This erratum occurs in the presence of following sequence of events

1. A memory access requiring VA to PA translation resulting in page table walk start.
2. For a translation stage when an *Input Address* (IA) is translated to an *Output Address* (OA), one of the following faults is received
   - Address size fault
   - Translation fault
   - Permission fault
3. The above conditions occur along with specific micro-architectural conditions with precise timing

Implications:

When ESR_ELx reports Address size, Translation or Permission fault, although software is expected to get no extra information from AFSR0_ELx, AFSR0_ELx.PORT can report "Overlapping illegal port" instead of "Unknown" and AFSR0_ELx.TYPE can report "Error on data" instead of "Other Error".

Workaround

No workaround is needed as AFSR0_ELx is not required to provide any useful information for Address size, Translation or Permission fault types.

3025593
Unaligned exclusive to Normal Non-cacheable memory is not aborted

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0 and r1p1. Fixed in r2p0.

Description

Cortex-R82 supports FEAT_LSE2 to allow unaligned exclusive instructions to be executed to Normal memory that is marked as both Inner Write-Back Cacheable and Outer Write-Back Cacheable. Unaligned exclusives to memory that is treated as Non-cacheable are not supported, but because of this erratum they are not aborted and the transactions will be sent on the *Main Manager* (MM) port or the *Low-Latency RAM* (LLRAM) port.

Configurations affected

This erratum affects all configurations of the Cortex-R82.

Conditions

This erratum occurs when an unaligned exclusive instruction is executed and both of the following conditions apply:

- It is to the MM port or the LLRAM port
- It is to Normal memory which does not have Inner Write-Back and Outer Write-Back Cacheable attributes
    - If the access is to LLRAM it also does not have Inner Write-Through and Outer Write-Through Cacheable attributes

Implications

When this erratum occurs and the exclusive is to the MM port, the exclusive will be sent on the MM port. The transactions sent on the MM port will be aligned. However the sizes might not match and the exclusive sequence may not succeed. If a store exclusive is sent out it will be an aligned transaction. However the transaction byte strobes will be set according to the size and address of the original store exclusive instruction. For exclusives targeting the LLRAM port, the transactions sent on the bus is sent as non-exclusive but the intention is that unaligned exclusives will abort.

There is no expectation that unaligned exclusives to Non-cacheable memory will succeed so software should not rely on these transactions.

Workaround

No workaround is required.

3105480
FAR register should record the lowest address that caused a fault

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1 and r2p0. Fixed in r3p0.

Description

Cortex-R82 accesses a load to 128-bit unaligned region as two separate aligned memory accesses, one to lower aligned address and other to higher aligned address. If any of these two receive *Data Abort exception*, its address is reported to *Fault Address Register*, *FAR_EL2 or FAR_EL1*, depending on current exception level. With this erratum, in case of *Data Abort exception* on higher aligned address, *Fault Address Register* still reports lower aligned address as the faulting address of the load to unaligned-128-bit region.

Configurations affected

This erratum affects all configurations of the Cortex-R82.

Conditions

This erratum occurs in the presence of following sequence of events:

1. A load instruction to 128-bit unaligned region is executed.
2. Lower aligned load of the unaligned 128-bit load completes without any fault.
3. Higher aligned load of the unaligned 128-bit load receives *Data Abort exception* due to an external error.
    - This could be due to an bus error, ECC error or poison

Implications

When this erratum occurs, *Fault Address Register* for the current exception level (i.e. *FAR_EL2 or FAR_EL1)* always reports lower aligned address as the faulting address from the two addresses associated with the load to unaligned-128-bit region on a *Data Abort exception*.

Workarounds

No workaround is required for this Erratum.

3114083
Cluster OFF_EMU to ON transition stuck because PACCEPT is blocked due to debug requests during OFF_EMU

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0 and r3p1. Open.

Description

The Cortex-R82 processor can use Emulated off (OFF_EMU) while powering down its cores to enable debug over powerdown sequences. The processor can then turn its cores ON directly from the OFF_EMU power mode.
Because of this erratum, if any core is transitioning from OFF_EMU to ON, core power transitions that are underway could deadlock.

Configurations affected

This erratum affects all configurations

Conditions

This erratum occurs when all the following conditions are met:

- At least one core is put in the Emulated off power mode
- An external debug (APB) request is made
- A core that is in Emulated off is requested to transition to ON
- Additional internal timing conditions are met

Implications

If this erratum occurs, outstanding power transitions could deadlock.

Workaround

To avoid this erratum, ensure that no external debug requests are active prior to requesting a core power transition to ON. If dynamic power transitions are enabled in the PPU, you must disable these by writing 0 to CLUSTERPPU_PWPR.PWR_DYN_EN before performing debug requests in OFF_EMU mode.

3123474
Some PMU events count incorrectly

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1 and r2p0. Fixed in r3p0.

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:

- 0x0074 ASE_SPEC
- 0x0075 VFP_SPEC
- 0x0391 DFB_RETIRED

Implications

If this erratum occurs:

- ASE_SPEC, VFP_SPEC or DFB_RETIRED might be over-counted. Software might not be able to use these events for performance analysis.

Workaround

There is no workaround.

3196425
Incorrect VA[14:12] sent to Instruction Cache for SnpDVMOp Virtual I-Cache Invalidate from CHI

Status

Affects: Cortex-R82
Fault type: Programmer Category C
Fault status: Present in r2p0. Fixed in r3p0.

Description

When Cortex-R82 *Main Manager* (MM) port is configured to use *Coherent Hub Interface* (CHI) and it is connected to a coherent cluster, it can receive a *Snoop DVM operation* (SnpDVMOp) from another cluster. With this Erratum, if the SnpDVMOp is of type *Virtual Instruction Cache Invalidate* (VICI), then virtual address bits extracted from VICI for the invalidation of Instruction Cache are partly from the wrong field of VICI.

Configurations affected

This erratum affects the configuration of the Cortex-R82 which has the MM port configured to be CHI.

Conditions

This erratum occurs when:

1. Cortex-R82 MM port, configured to be CHI, is connected to another coherent non-Cortex-R82 cluster through a coherency interconnect
2. A SnpDVMOp of type VICI is received by Cortex-R82

Implications

There is no implication of this Erratum if code is shared only between Cortex-R82 clusters. This is because Cortex-R82 cores do not use these transactions for I-Cache instruction invalidation.

If code is shared between a Cortex-R82 cluster and another coherent non-Cortex-R82 cluster that can generate SnpDVMOp-VICI, then this Erratum has the implication of invalidating I-Cache locations using *payload ASID[2:0]* as *VA[14:12]*. This will result in not invalidating the targeted I-Cache locations, potentially causing the core to execute stale instructions.

Workaround

No workaround is required if code is shared only between Cortex-R82 clusters. If the code is shared with another coherent non-Cortex-R82 cluster that can generate SnpDVMOp-VICI, then the only workaround is for this other cluster to use SnpDVMOp of types other than VICI for I-Cache invalidation.

3197529
AT operation might incorrectly indicate a Synchronous External abort in PAR_EL1

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1 and r2p0. Fixed in r3p0.

Description

The Cortex-R82 processor supports multiple memory ports and any memory accesses that target a port that is currently disabled, or where multiple ports overlap, will return a Synchronous External abort. There are also implementation defined controls to restrict some accesses that might impact interrupt latency and if a prohibited access is attempted it will generate a Synchronous External abort. Due to this erratum an AT operation that meets the condition of one of these implementation defined aborts might incorrectly report a Synchronous External abort fault occurred in the value of the PAR_EL1 register.

Configurations affected

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

Conditions

This erratum occurs when the following sequence of events occurs:

1. An AT operation is performed
2. No translation fault occurs for the AT operation
3. The address of the AT operation meets one of the following conditions:
    - The address to be translated is to a port that is disabled
    - The address to be translated corresponds to multiple overlapping ports
    - The address has Device attributes and IMP_INTLATENCY_EL2.DEV == 0b1

Implications

When this erratum occurs, the result of the AT operation will be written into the PAR_EL1 register with PAR_EL1.F == 0b1 and PAR_EL1.FST == 0b010000.

Workaround

No workaround is required for this erratum.

3203547
IMP_INTLATENCY_EL2.DEV might cause a Synchronous External abort for IC instructions to Device addresses

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1 and r2p0. Fixed in r3p0.

Description

In order to guarantee interrupt latency, the Cortex-R82 processor provides implementation defined controls to restrict certain operations. IMP_INTLATENCY_EL2.DEV allows restricting Device access to the *Main Manager* (MM) or *Low-latency RAM* (LLRAM) port, and restricts transactions to the *Shared Peripheral Port* (SPP) or *Low-latency Peripheral Port* (LLPP). This also restricts transactions that would result in multiple bus transactions to the SPP or LLPP.

Configurations affected

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

Conditions

This erratum occurs when the following sequence of events occurs:

1. IMP_INTLATENCY_EL2.DEV is set to 1
2. An IC IVAU instruction is executed, with the address meeting the conditions for IMP_INTLATENCY_EL2.DEV to cause a Synchronous External abort

Implications

When this erratum occurs, the IC operation will take a Synchronous External abort. It is not expected that software will attempt instruction cache maintenance to Device addresses and so the conditions for this erratum should not be likely to occur.

Workaround

No workaround is required for this erratum.

3338838
IMP_CLUSTERACTLR_EL1.L2SPEC specification is incorrect

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

The IMP_CLUSTERACTLR_EL1.L2SPEC register bit which is documented as controlling L2 cache speculative accesses can be read and written, but has no effect on the processor

Configurations affected

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

Conditions

This erratum occurs when the following condition takes place:

- The IMP_CLUSTERACTLR_EL1.L2SPEC register is written with a non-zero value

Implications

L2 cache speculative accesses are still enabled.

Workaround

There is no workaround. IMP_CLUSTERACTLR_EL1.L2SPEC should be written as zero.

3378740
ECC error in L1 D-cache tag RAM might not get logged in RAS record

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

The Cortex-R82 processor supports reporting errors via the *Reliability, Availability, and Serviceability* (RAS) registers. Due to this erratum, it is possible that a corrected error in the L1 data cache Tag RAM is not reported.

Configurations affected

This erratum affects all configurations of the Cortex-R82 processor with the RAM_PROTECTION configuration parameter set to 1.

Conditions

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

1. An error has previously occurred in the L1 data cache, in either the data RAM or the tag RAM
2. A cacheline containing data for the *Low-latency RAM* (LLRAM) port is held in the L1 data cache
3. A linefill is performed for a *Main Manager* (MM) address, for example due to a load or store instruction or a pagewalk, which uses the same set and way as the LLRAM cacheline
4. A cache lookup occurs to the same set as the previous error, for example due to a load or store instruction or a pagewalk
5. An new error occurs in the L1 data cache tag RAM for the set and way of the linefill that was started for the MM address in condition 3, with no other cache lookups being performed

Implications

When this erratum occurs, the Tag RAM contents will be updated with the correct data for the new cacheline, but the correction of the error will not be reported in the RAS registers.

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

Workaround

No workaround is required for this erratum.

3454202
Unused utility bus address locations are aliased

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

This erratum affects the default sparse address map mode. If a utility bus access is performed to an address in an unused 4kB address region, the access should be treated as RAZ/WI. Instead, the the access will be aliased onto an existing register location.

Configurations affected

This erratum affects the configuration with sparse address map (DENSE_CS_ADDR_MAP = 0).

Conditions

This erratum occurs when all the following conditions are met:

- A utility bus access is performed with non-zero offset address in the bit position 15 to 12.

Implications

If this erratum occurs, some external registers may be read or modified unintentionally. The register access restrictions for the alias locations are respected.

Workaround

No workaround is required.

3545298
An external Non-cacheable store exclusive receiving a bus error response might not raise an asynchronous abort

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0 and r3p1. Open.

Description

The Cortex-R82 processor supports sending Non-cacheable or Device exclusives on the *Main Manager* (MM) and *Low-Latency RAM* (LLRAM) ports. If a Non-cacheable or Device store exclusive gets an error response, the store exclusive might only abort synchronously when it cannot be guaranteed that memory has not been updated.

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. An store exclusive instruction is executed with Non-cacheable or Device attributes
2. The store exclusive is sent on the MM or LLRAM port
3. The response to the store exclusive is an error response
    - For AXI, BRESP has a value of SLVERR or DECERR
    - For CHI, the response is a DErr or NDErr response
4. The memory targeted by the store exclusive is still modified despite the previous error occurring

Implications

If this erratum occurs a synchronous exception will be raised despite the associated memory location having been updated.

Workaround

No workaround is expected to be needed as it is expected that for most systems Non-cacheable and Device exclusives are not used, also this erratum can only occur when there is a bus error which might not be possible to recover from.

3606619
L2 Flushes too many ways when moving from FULL_RAM to HALF_RAM

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

The Cortex-R82 processor supports the power down of half of the L2 cache in order to save power. Maintaining coherency and data consistency is achieved by evicting half of the data from the L2 cache before powering off the respective half of the L2 cache RAM. Due to this erratum, all the lines in the L2 cache are evicted rather than only the lines that are needed.

Configurations affected

This erratum affects configurations of the Cortex-R82 processor where the L2 cache size is greater than zero.

Conditions

A power transition occurs from FULL_RAM to HALF_RAM operating mode.

Implications

All lines in the L2 cache are flushed rather than just the cachelines for the half of the cache being powered down. This can result in a performance loss due to the need to refetch some cache lines from main memory after the power mode transition.

Workaround

There is no workaround.

3639153
Cold debug recovery reset will not reset some system registers

Status

Affects: Cortex-R82
Fault type: Programmer Category C
Fault status: Present in r1p0, r1p1, r2p0, r3p0 and r3p1. Open.

Description

When a cold debug recovery reset is applied to the Cortex-R82 processor system, there are some system registers that will not be reset to their initial value.

Configurations affected

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

Conditions

This erratum can occur when the following sequence of events occurs

1. Cluster power transition from OFF or MEM_RET to ON
2. CLUSTERPPU_PTCR.DBG_RECOV_PORST_EN has been programmed to be 1'b1
3. Cluster power transition from ON to DBG_RECOV
4. Cluster power transition from DBG_RECOV back to ON

Implications

When any cold debug recovery reset is applied, the following system register fields will not be reset to their corresponding initial value.

- IMP_CLUSTERACTLR_EL1.UBACCLVL
- IMP_CLUSTERACELSCTLR_EL1.LLRAMACCLVL
- IMP_CLUSTERACELSCTLR_EL1.TCMACCLVL<n>, where n is from 0 to NUM_CORES - 1
- IMP_CLUSTERPWRCTLR_EL1.PRTNRQ
- IMP_CLUSTERPWRDN_EL1.MEM_RET

Workaround

Arm does not expect a workaround to be required for this erratum.

3683909
The CPU ROM Table reports an incorrect base address for CTI2-7 for configurations with DENSE_CS_ADDR_MAP = 1

Status

Affects: Cortex-R82
Fault type: Programmer Category C
Fault status: Present in r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

The CPU ROM Table reports an incorrect base address for CTI2-7 in configurations with DENSE_CS_ADDR_MAP = 1.

Configurations affected

The erratum affects configurations of the Cortex-R82 processor with the configuration parameter DENSE_CS_ADDR_MAP = 1.

Conditions

This erratum can occur when we read out the Core <n> CTI component address from DBROM_ROMENTRY<n + 2>, where n = 2 to 7.

Implications

Incorrect CTI component address is read out for cores 2 to 7 from the DBROM_ROMENTRY registers. If the incorrect address is accessed the registers will be treated as RAZ/WI.

Workaround

The CTI component registers can be correctly accessed at the base addresses specified in the Cortex-R82 Technical Reference Manual.

3686657
Interrupt might be taken on a read from device memory that receives a bus error

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

When a read from device memory is performed that receives a response indicating a bus error, it is possible for an asynchronous interrupt to be taken instead of the synchronous exception for the data abort.

Configurations affected

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

Conditions

This erratum occurs when the following sequence of events occurs:

1. Interrupts are not masked
2. A read to device memory is performed
3. An interrupt is received before the response to the read from device memory
    - The interrupt can be a IRQ/FIQ/SEI/vIRQ/vFIQ/vSEI interrupt
4. The response to the device read has a bus error

Implications

An interrupt might be taken instead of the synchronous data abort for the device read. If the return address is not modified in the interrupt handler, on executing an ERET the device load will be repeated and take the data abort exception as long as no interrupt is pending and the response returns the same bus error. The original synchronous data abort on a read affected by this erratum will be lost unless the issue that resulted in the bus error is recorded in the system.

Workaround

There is no workaround for this erratum.

3769700
L2 incorrectly applies address-based hazarding to Non-Reorderable Device writes, reducing performance

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0 and r3p0. Fixed in r3p1.

Description

The L2 memory system incorrectly applies address based hazarding to write requests targeting Non-Reorderable Device memory, reducing the throughput of write requests.

Configurations affected

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

Conditions

This erratum occurs when the processor executes a write targeting Non-Reorderable Device memory on the *Main Manager* (MM) port.

Implications

The throughput of Non-Reorderable Device writes targeting the L2 memory system is reduced. When the processor is configured with L2_SLICES=2 Non-Reorderable Device writes for which the address bit [6] == 0 might incur additional hazarding compared to Non-Reorderable Device writes with address bit [6] == 1.

Workaround

No workaround is expected to be required for this erratum.

3776502
L2 incorrectly applies address-based hazarding to Reorderable Device writes, reducing performance

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0 and r3p1. Open

Description

The L2 memory system incorrectly applies address based hazarding to write requests targeting Reorderable Device memory, reducing the throughput of write requests.

Configurations affected

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

Conditions

This erratum occurs when the processor executes a write targeting Reorderable Device memory on the *Main Manager* (MM) port.

Implications

The throughput of Reorderable Device writes targeting the L2 memory system is reduced.

Workaround

No workaround is expected to be required for this erratum.

3823246
Cluster PMU does not generate overflow

Status

Affects: Cortex-R82
Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r1p0, r1p1, r2p0, r3p0 and r3p1. Open

Description

The Cortex-R82 processor contains a PMU that counts events in the shared cluster logic. Due to this erratum, overflows of event or cycle counts in the cluster PMU will not be recorded.

Configurations affected

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

Conditions

This erratum occurs when a cluster PMU counter overflows.

Implications

A read of the IMP_CLUSTERPMOVSSET_EL1 and IMP_CLUSTERPMOVSCLR_EL1 will not reflect if a cluster PMU counter has overflowed. The nCLUSTERPMUIRQ interrupt will not be asserted on an overflow of a cluster PMU counter, but is asserted if a write to IMP_CLUSTERPMOVSSET_EL1 causes the overflow flag to be set. Cluster PMU counters are 32-bits, except for the cycle counter which is 64-bits.

When the Cluster PMU counters are being used as free running counters the smallest time for an overflow to occur in a counter when running at a clock frequency of 1GHz is greater than 500ms. The CHAIN event can be used to treat two of the 32-bit counters as a 64-bit counter which will multiply the minimum time for an overflow to occur by 2^32. The minimum time in ms for an overflow to occur for the 32-bit counters can be calculated by `time = 536 / <frequency in GHz>`.

When overflows need to be tracked in the cluster PMU software should check the count values periodically, more frequently then the minimum time to overflow. If the count value is observed to decrease compared to the previous value this implies an overflow has occurred and software can manually record this by writing to the IMP_CLUSTERPMOVSSET_EL1 register.

Workaround

No workaround is expected to be required for this erratum.

Proprietary notice

(PRE-1121-V1.0)

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Product revision status

The rxpy identifier indicates the revision status of the product described in this manual, where:

rx

Identifies the major revision of the product.

py

Identifies the minor revision or modification status of the product.