arm

Cortex-M85 AT640 and Cortex-M85 with FPU AT641

Software Developer Errata Notice

Date of issue: April 16, 2024
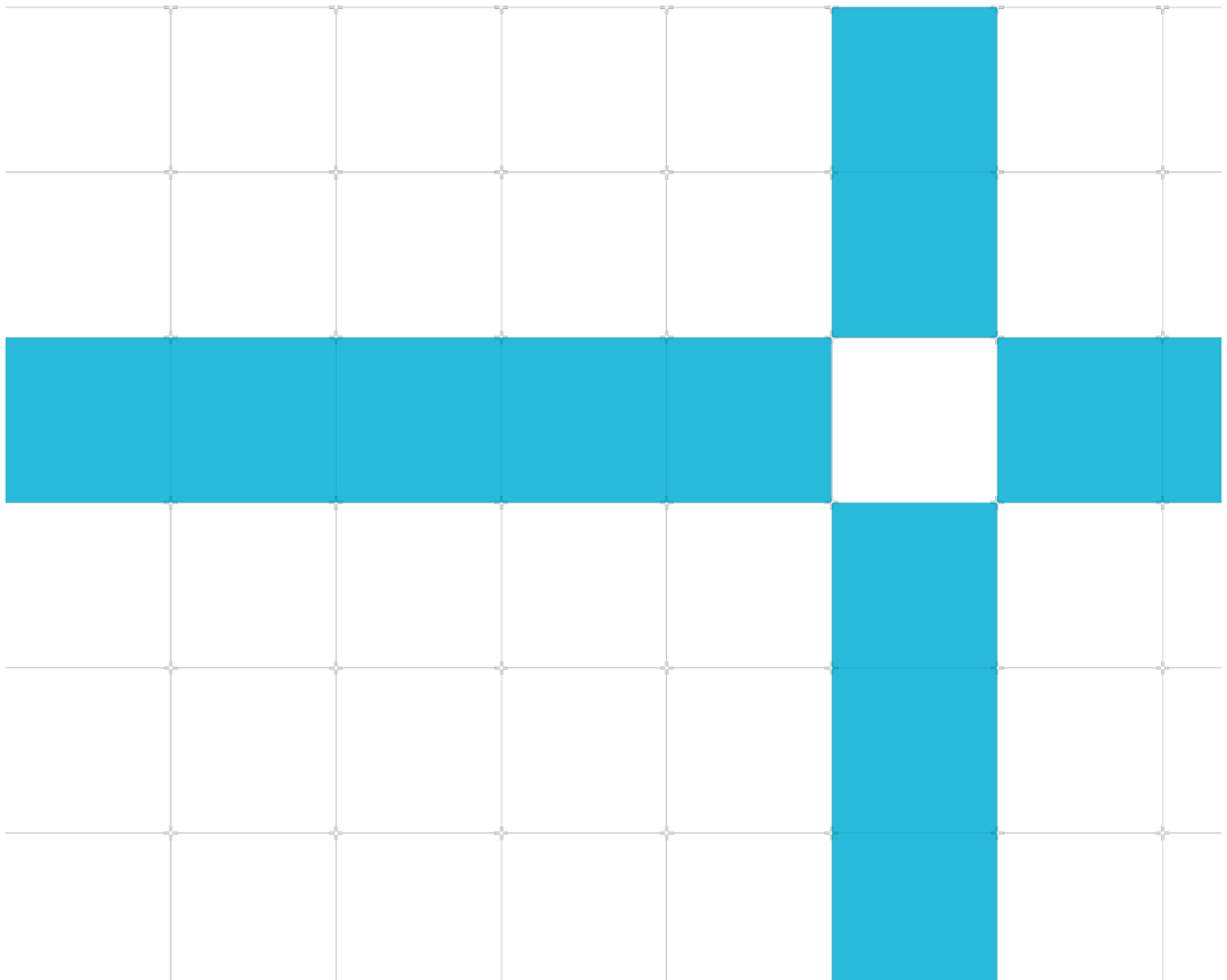
Document version: 14.0

Document ID: SDEN-2236668

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

Non-confidential proprietary notice

Confidentiality status

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

Unrestricted Access is an Arm internal classification.

Product status

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Cortex-M85 AT640 and Cortex-M85 with FPU AT641, create a ticket on **https://support.developer.arm.com**.

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

Inclusive language commitment

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

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

Contents

r0p0 implementation fixes

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

REVIDR[0]	2286936 Aliased branch prediction in an IT block near an exception entry or BLXNS may result in incorrect execution
REVIDR[1]	2355499 Under limited circumstances, write-through store operations do not update the data cache correctly

Note that there is no change to the CPUID which remains at r0p0. Software will identify this release through the combination of CPUID and REVIDR.

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.

April 16, 2024: Changes in document version v14.0

ID	Status	Area	Category	Summary
2958637	Updated	Programmer	Category B	Certain PMU EVENTBUS events not driven correctly on M85
3175626	Updated	Programmer	Category B	AXI hang due to dependency between read data channel and write response channel
3190818	Updated	Programmer	Category B	Under limited circumstances, LDM to normal non-cacheable AXI location can not complete
3319763	New	Programmer	Category C	ACTLR_S bit[11] can not disable No Write-Allocate mode

March 07, 2024: Changes in document version v13.0

ID	Status	Area	Category	Summary
2627033	Updated	Programmer	Category B	An LDR following a flushed LDM targeting the same Non-cacheable line can return corrupted data
3175626	Updated	Programmer	Category B	AXI hang due to dependency between read data channel and write response channel
3190818	Updated	Programmer	Category B	Under limited circumstances, LDM to normal non-cacheable AXI location can not complete
2705514	Updated	Programmer	Category B (rare)	Under limited circumstances, an LDM following a flushed LDM from Non-cacheable region can result in deadlock
2282154	Updated	Programmer	Category C	Under limited circumstances, a load to a normal non-cacheable location might not be properly ordered by a barrier instruction

February 07, 2024: Changes in document version v12.0

ID	Status	Area	Category	Summary
3175626	New	Programmer	Category B	AXI hang due to dependency between read data channel and write response channel
3190818	New	Programmer	Category B	Under limited circumstances, LDM to normal non-cacheable AXI location can not complete

November 30, 2023: Changes in document version v11.0

ID	Status	Area	Category	Summary
3078268	New	Programmer	Category C	Incorrect multi-copy atomicity ordering between Non-cacheable loads with forwarded store data

September 01, 2023: Changes in document version v10.0

ID	Status	Area	Category	Summary
2958637	New	Programmer	Category B	Certain PMU EVENTBUS events not driven correctly on M85
2902253	New	Programmer	Category C	ITM can generate synchronization and timestamp packets when NIDEN=0
2988396	New	Programmer	Category C	DWT data value packets can be generated while the match packet from same access overflows
3030982	New	Programmer	Category C	Store data value for certain aligned vector store incorrect to DWT for data value matching

November 23, 2022: Changes in document version v9.0

ID	Status	Area	Category	Summary
2722045	Updated	Programmer	Category B	Under certain conditions, a load or store crossing into an MPU UNDEFINED region can report an unaligned UsageFault instead of the expected MemManage Fault
2705514	Updated	Programmer	Category B (rare)	Under limited circumstances, an LDM following a flushed LDM from Non-cacheable region can result in deadlock
2640876	Updated	Programmer	Category C	Load data value for certain unaligned vector loads incorrect to DWT for data value matching
2651727	Updated	Programmer	Category C	Unprivileged debug transactions to the STIR and EVENTSPR registers can pend interrupts
2666660	Updated	Programmer	Category C	An unusual combination of events can cause the core to hang instead of lock up on SOE entry
2666664	Updated	Programmer	Category C	Under limited conditions, an unaligned MVE load reading the TCMs may return corrupted data if it encounters a transient BusFault on its TCM read
2674410	Updated	Programmer	Category C	An AXI load reading some portion of a word that an older store is writing might fail to report a BusFault in certain situations
2677972	Updated	Programmer	Category C	CTITRIGIN not connected to DWT_CMPMATCH for ETM = 0 configuration
2682779	Updated	Programmer	Category C	After deactivating the instruction cache, self-modified code might not be executed correctly
2705117	Updated	Programmer	Category C	L1D_CACHE_REFILL and L1D_CACHE_MISS_RD PMU events might be inaccurate
2746551	New	Programmer	Category C	Store address to DWT incorrect for certain VSTR operations
2757159	New	Programmer	Category C	Under limited circumstances, store operations do not update the data cache correctly
2444277	Updated	Programmer	Category C	Store data value for certain MVE stores is incorrect to DWT for data value matching
2459389	Updated	Programmer	Category C	Executing VSCCLRM with an invalid ICI value may raise INVSTATE UsageFault

September 06, 2022: Changes in document version v8.0

ID	Status	Area	Category	Summary
2722045	New	Programmer	Category B	Under certain conditions, a load or store crossing into an MPU UNDEFINED region can report an unaligned UsageFault instead of the expected MemManage Fault
2705514	New	Programmer	Category B (rare)	Under limited circumstances, an LDM following a flushed LDM from Non-cacheable region can result in deadlock
2640876	New	Programmer	Category C	Load data value for certain unaligned vector loads incorrect to DWT for data value matching
2651727	New	Programmer	Category C	Unprivileged debug transactions to the STIR and EVENTSPR registers can pend interrupts
2666660	New	Programmer	Category C	An unusual combination of events can cause the core to hang instead of lock up on SOE entry
2666664	New	Programmer	Category C	Under limited conditions, an unaligned MVE load reading the TCMs may return corrupted data if it encounters a transient BusFault on its TCM read
2674410	New	Programmer	Category C	An AXI load reading some portion of a word that an older store is writing might fail to report a BusFault in certain situations
2677972	New	Programmer	Category C	CTITRIGIN not connected to DWT_CMPMATCH for ETM = 0 configuration
2682779	New	Programmer	Category C	After deactivating the instruction cache, self-modified code might not be executed correctly
2705117	New	Programmer	Category C	L1D_CACHE_REFILL and L1D_CACHE_MISS_RD PMU events might be inaccurate
2444277	New	Programmer	Category C	Store data value for certain MVE stores is incorrect to DWT for data value matching

April 27, 2022: Changes in document version v7.0

ID	Status	Area	Category	Summary
2615656	New	Programmer	Category A	Under certain conditions, an unaligned MVE load can return incorrect data for the lower word in its second tick if an older store is writing the same word
2467111	Updated	Programmer	Category A	Conditional LCTP may use out of date flag values for bypass or update of LTPSIZE
2627033	New	Programmer	Category B	An LDR following a flushed LDM targeting the same Non-cacheable line can return corrupted data
2257706	Updated	Programmer	Category B	Under limited circumstances, store operations do not update the data cache correctly
2296335	Updated	Programmer	Category B	DWT Watchpoints might not work reliably on Data address and Data value comparator matches
2312674	Updated	Programmer	Category B	Under limited circumstances, PLDs to the same cache line address as pending line-fills cause duplicate cache allocation and thus data corruption

ID	Status	Area	Category	Summary
2425308	Updated	Programmer	Category B	When an MVE store encounters a fault, in some cases the FAR is not updated to the address of the beat that reports the fault
2449355	Updated	Programmer	Category B	A word-crossing store which gets an ignored BusFault for one of its words can cause a hang when dual-issued with a younger Device load
2485388	New	Programmer	Category B	An ongoing DSB does not wait for the automatic invalidation of the data cache to complete before retiring
2434209	Updated	Programmer	Category B (rare)	Certain stall cases can cause an unaligned MVE load to incorrectly zero some of its return data or fail to report a fault
2490920	New	Programmer	Category C	A malformed tail-predicated loop at the start of a Secure function may use the value of LTPSIZE without masking
2589246	New	Programmer	Category C	An incorrect fault address is reported for an unprivileged aligned doubleword store to the STIR and RFSR registers when CCR.USERSETMPEND is set
2593278	New	Programmer	Category C	Under limited conditions, an imprecise abort may not be reported when a tag RAM ECC error is detected and data loss has occurred
2601889	New	Programmer	Category C	Under limited circumstances, an ECC error on data cache RAMs can lead to data corruption
2614056	New	Programmer	Category C	Under limited conditions, a doubleword-aligned doubleword ITCM load can read data corrupted by an ECC error
2631240	New	Programmer	Category C	Unexpected writes to DEBRs can potentially stop the outstanding linefills from completion
2266957	Updated	Programmer	Category C	Incorrect execution priority may be applied for a short duration in rare cases after the start of an IRQ return
2374351	Updated	Programmer	Category C	Trace fails to update PC on Imprecise bus fault under IESB fault escalation
2388605	Updated	Programmer	Category C	A cacheable load which misses the data cache due to a correctable tag ECC error can return stale data if the cacheline it is reading is dirty
2397109	Updated	Programmer	Category C	An Instruction might be incorrectly sent to Embedded Trace Macrocell as partially completed instruction
2432946	Updated	Programmer	Category C	ETM reports wrong address on tail-chain during sleep on exit
2434207	Updated	Programmer	Category C	An MVE tick that encounters faults in both beats may report the FAR address incorrectly if the fault types for each beat are different
2434651	Updated	Programmer	Category C	An aligned doubleword MVE load reading the ITCM with one word predicated out might read data corrupted by an ECC error for the non-predicated word
2439477	Updated	Programmer	Category C	A Secure read of ICSR_NS.VECTPENDING might return the Secure value
2445488	Updated	Programmer	Category C	ETM incorrectly reports there are also additional serious exceptions when reporting entry into a serious exception
2446484	Updated	Programmer	Category C	Simultaneous ECC errors on a store tag lookup and a load data read in the data cache can cause the ECC error for the load to not get corrected

ID	Status	Area	Category	Summary
2451384	Updated	Programmer	Category C	CHAIN PMU event unconnected to PMU EVENTBUS
2452728	New	Programmer	Category C	Before RAM power is up and completes cache auto invalidation sequence, i.e. D$ is accessible, a cacheable store data might not be observed by a load to the same address after D$ is accessible
2456913	Updated	Programmer	Category C	Under limited circumstances, an LDM to a normal non-cacheable location followed by store to the same location might prevent the core from transitioning to lower power mode
2467154	New	Programmer	Category C	S-AHB read bursts might terminate prematurely in case TCM Error is encountered by a speculatively prefetched read which immediately follows a read that encounters correctable ECC error

March 18, 2022: Changes in document version v6.0

ID	Status	Area	Category	Summary
2467111	New	Programmer	Category A	Conditional LCTP may use out of date flag values for bypass or update of LTPSIZE
2296335	Updated	Programmer	Category B	DWT Watchpoints might not work reliably on Data address and Data value comparator matches
2312674	Updated	Programmer	Category B	Under limited circumstances, PLDs to the same cache line address as pending line-fills cause duplicate cache allocation and thus data corruption
2425308	New	Programmer	Category B	When an MVE store encounters a fault, in some cases the FAR is not updated to the address of the beat that reports the fault
2449355	New	Programmer	Category B	A word-crossing store which gets an ignored BusFault for one of its words can cause a hang when dual-issued with a younger Device load
2434209	New	Programmer	Category B (rare)	Certain stall cases can cause an unaligned MVE load to incorrectly zero some of its return data or fail to report a fault
2432946	New	Programmer	Category C	ETM reports wrong address on tail-chain during sleep on exit
2434207	New	Programmer	Category C	An MVE tick that encounters faults in both beats may report the FAR address incorrectly if the fault types for each beat are different
2434651	New	Programmer	Category C	An aligned doubleword MVE load reading the ITCM with one word predicated out might read data corrupted by an ECC error for the non-predicated word
2439477	New	Programmer	Category C	A Secure read of ICSR_NS.VECTPENDING might return the Secure value
2445488	New	Programmer	Category C	ETM incorrectly reports there are also additional serious exceptions when reporting entry into a serious exception
2446484	New	Programmer	Category C	Simultaneous ECC errors on a store tag lookup and a load data read in the data cache can cause the ECC error for the load to not get corrected
2451384	New	Programmer	Category C	CHAIN PMU event unconnected to PMU EVENTBUS
2456913	New	Programmer	Category C	Under limited circumstances, an LDM to a normal non-cacheable location followed by store to the same location might prevent the core from transitioning to lower power mode
2459389	New	Programmer	Category C	Executing VSCCLRM with an invalid ICI value may raise INVSTATE UsageFault

January 21, 2022: Changes in document version v5.0

ID	Status	Area	Category	Summary
2266957	Updated	Programmer	Category C	Incorrect execution priority may be applied for a short duration in rare cases after the start of an IRQ return
2388605	New	Programmer	Category C	A cacheable load which misses the data cache due to a correctable tag ECC error can return stale data if the cacheline it is reading is dirty
2397109	New	Programmer	Category C	An Instruction might be incorrectly sent to Embedded Trace Macrocell as partially completed instruction

November 29, 2021: Changes in document version v4.0

ID	Status	Area	Category	Summary
2295129	New	Programmer	Category B	From an unprivileged debugger, cache maintenance and ERRDEVID registers are not accessible while ERRIIDR is accessible
2371188	New	Programmer	Category B	A flag writing external coprocessor instruction may cause incorrect bypass from younger conditional instructions
2355269	New	Programmer	Category C	A DWORD-sized store access to ITM_CIDR2 and ITM_CIDR3 can fault incorrectly on the access to the ITM_CIDR3 when the store is not privileged
2365142	New	Programmer	Category C	Direct cache access to Instruction cache data RAM cannot read half of the RAM locations
2371473	New	Programmer	Category C	The execution priority might be incorrect for a cycle when an IRQ's priority is updated in its handler
2374006	New	Programmer	Category C	Undefined 16bit opcodes could raise a NOCP UsageFault instead of an UNDEFINSTR UsageFault
2374269	New	Programmer	Category C	Unprivileged debugger access allowed to read or write the DPDLPSTATE register
2374351	New	Programmer	Category C	Trace fails to update PC on Imprecise bus fault under IESB fault escalation
2374615	New	Programmer	Category C	FP exception flags in the FPSCR might be set incorrectly after FP context creation following a security state change

November 11, 2021: Changes in document version v3.0

ID	Status	Area	Category	Summary
2355499	New	Programmer	Category A	Under limited circumstances, write-through store operations do not update the data cache correctly
2288773	New	Programmer	Category B	A VLDRD executing in big-endian mode returns incorrect data if the load reads a word from device memory that is fully predicated when the other word isn't fully predicated
2288808	New	Programmer	Category B	Under limited circumstances, loads do not properly hazard other cache updates which cause a duplicate entry to be placed in the data cache causing data corruption
2296335	New	Programmer	Category B	DWT Watchpoints might not work reliably on Data address and Data value comparator matches
2309351	New	Programmer	Category B	Incorrect values may be read from FP registers and VPR
2312674	New	Programmer	Category B	Under limited circumstances, PLDs to the same cache line address as pending line-fills cause duplicate cache allocation and thus data corruption
2315530	New	Programmer	Category B	IESB derived fault hang during entry
2338043	New	Programmer	Category B	The fields IDC, IXC, UFC, OFC and IOC in the FPSCR might be corrupted after the execution of double precision floating point multiply and accumulate instruction
2272772	New	Programmer	Category C	Debug read to FPSCR may return wrong default value
2276413	New	Programmer	Category C	Debug writes to privleged only bits of current state using DCRSR may not occur if UDE is enabled
2282154	New	Programmer	Category C	Under limited circumstances, a load to a normal non-cacheable location might not be properly ordered by a barrier instruction
2283939	New	Programmer	Category C	TCM requests that receive both a TGU fault and have forwarded data do not properly report a TGU fault
2287723	New	Programmer	Category C	The vector register file may not be cleared when tail chaining from a secure to non-secure exception
2297046	New	Programmer	Category C	RTL incorrectly halts as it incorrectly computes vector catch on lockup entry
2311453	New	Programmer	Category C	Flaw in memory system power down may result in data corruption
2312664	New	Programmer	Category C	Under limited circumstances, LDRD or VLDR instructions that access both device and non-device memory can deadlock the memory system
2319196	New	Programmer	Category C	DWT_PCSR can present next non-committed instruction when no instruction retirement occurs in the core
2321988	New	Programmer	Category C	LSERR v/s NOCP Fault prioritization during PushStack
2323280	New	Programmer	Category C	Word crossing store accesses that access the last word of the PAHB memory region might cause data corruption or incorrect fault behavior
2335473	New	Programmer	Category C	A load or store multiple might cause an incorrect ICI value to be stacked in the RETPSR

October 13, 2021: Changes in document version v2.0

ID	Status	Area	Category	Summary
2286936	New	Programmer	Category A	Aliased branch prediction in an IT block near an exception entry or BLXNS may result in incorrect execution
2247184	New	Programmer	Category B	Automatic EWIC register save/restore sequence in some rare cases disables EWIC capability of recognizing wakeup events
2257706	New	Programmer	Category B	Under limited circumstances, store operations do not update the data cache correctly
2266165	New	Programmer	Category B	An ignored bus fault can cause an operation to not make forward progress
2242544	New	Programmer	Category C	Data corruption may be observed if there are consecutive reads to different Wakeup Event Mask registers
2253502	New	Programmer	Category C	An interstating LDM may restart using an incorrect base address
2253509	New	Programmer	Category C	A debug monitor step may cause an incorrect ICI value to be stacked in the RETPSR
2265059	New	Programmer	Category C	A store with memory attributes that do not match a younger LDM to the same address may cause a hang
2266957	New	Programmer	Category C	Incorrect execution priority may be applied for a short duration in rare cases after the start of an IRQ return
2279775	New	Programmer	Category C	An UNDEFINSTR fault could be prioritized over an INVSTATE fault for some invalid ICI values

June 29, 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
2286936	Programmer	Category A	Aliased branch prediction in an IT block near an exception entry or BLXNS may result in incorrect execution	r0p0	r0p1
2355499	Programmer	Category A	Under limited circumstances, write-through store operations do not update the data cache correctly	r0p0	r0p1
2615656	Programmer	Category A	Under certain conditions, an unaligned MVE load can return incorrect data for the lower word in its second tick if an older store is writing the same word	r0p0, r0p1	r0p2
2467111	Programmer	Category A	Conditional LCTP may use out of date flag values for bypass or update of LTPSIZE	r0p0, r0p1	r0p2
2247184	Programmer	Category B	Automatic EWIC register save/restore sequence in some rare cases disables EWIC capability of recognizing wakeup events	r0p0	r0p1
2257706	Programmer	Category B	Under limited circumstances, store operations do not update the data cache correctly	r0p0, r0p1	r0p2
2266165	Programmer	Category B	An ignored bus fault can cause an operation to not make forward progress	r0p0	r0p1
2485388	Programmer	Category B	An ongoing DSB does not wait for the automatic invalidation of the data cache to complete before retiring	r0p0, r0p1	r0p2
2338043	Programmer	Category B	The fields IDC, IXC, UFC, OFC and IOC in the FPSCR might be corrupted after the execution of double precision floating point multiply and accumulate instruction	r0p0	r0p1
2312674	Programmer	Category B	Under limited circumstances, PLDs to the same cache line address as pending line-fills cause duplicate cache allocation and thus data corruption	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2309351	Programmer	Category B	Incorrect values may be read from FP registers and VPR	r0p0	r0p1
2296335	Programmer	Category B	DWT Watchpoints might not work reliably on Data address and Data value comparator matches	r0p0, r0p1	r0p2
2288808	Programmer	Category B	Under limited circumstances, loads do not properly hazard other cache updates which cause a duplicate entry to be placed in the data cache causing data corruption	r0p0	r0p1
2295129	Programmer	Category B	From an unprivileged debugger, cache maintenance and ERRDEVID registers are not accessible while ERRIIDR is accessible	r0p0	r0p1
2288773	Programmer	Category B	A VLDRD executing in big-endian mode returns incorrect data if the load reads a word from device memory that is fully predicated when the other word isn't fully predicated	r0p0	r0p1
2315530	Programmer	Category B	IESB derived fault hang during entry	r0p0	r0p1
2425308	Programmer	Category B	When an MVE store encounters a fault, in some cases the FAR is not updated to the address of the beat that reports the fault	r0p0, r0p1	r0p2
2371188	Programmer	Category B	A flag writing external coprocessor instruction may cause incorrect bypass from younger conditional instructions	r0p0	r0p1
2627033	Programmer	Category B	An LDR following a flushed LDM targeting the same Non-cacheable line can return corrupted data	r0p0, r0p1	r0p2
2722045	Programmer	Category B	Under certain conditions, a load or store crossing into an MPU UNDEFINED region can report an unaligned UsageFault instead of the expected MemManage Fault	r0p0, r0p1, r0p2	r1p0
2449355	Programmer	Category B	A word-crossing store which gets an ignored BusFault for one of its words can cause a hang when dual-issued with a younger Device load	r0p0, r0p1	r0p2

ID	Area	Category	Summary	Found in versions	Fixed in version
3190818	Programmer	Category B	Under limited circumstances, LDM to normal non-cacheable AXI location can not complete	r0p0, r0p1, r0p2, r1p0	r1p1
2958637	Programmer	Category B	Certain PMU EVENTBUS events not driven correctly on M85	r0p0, r0p1, r0p2, r1p0	r1p1
3175626	Programmer	Category B	AXI hang due to dependency between read data channel and write response channel	r0p0, r0p1, r0p2, r1p0	r1p1
2434209	Programmer	Category B (rare)	Certain stall cases can cause an unaligned MVE load to incorrectly zero some of its return data or fail to report a fault	r0p0, r0p1	r0p2
2705514	Programmer	Category B (rare)	Under limited circumstances, an LDM following a flushed LDM from Non-cacheable region can result in deadlock	r0p0, r0p1, r0p2	r1p0
2242544	Programmer	Category C	Data corruption may be observed if there are consecutive reads to different Wakeup Event Mask registers	r0p0	r0p1
2434651	Programmer	Category C	An aligned doubleword MVE load reading the ITCM with one word predicated out might read data corrupted by an ECC error for the non-predicated word	r0p1	r0p2
2434207	Programmer	Category C	An MVE tick that encounters faults in both beats may report the FAR address incorrectly if the fault types for each beat are different	r0p0, r0p1	r0p2
2432946	Programmer	Category C	ETM reports wrong address on tail-chain during sleep on exit	r0p0, r0p1	r0p2
2266957	Programmer	Category C	Incorrect execution priority may be applied for a short duration in rare cases after the start of an IRQ return	r0p0, r0p1	r0p2
2265059	Programmer	Category C	A store with memory attributes that do not match a younger LDM to the same address may cause a hang	r0p0	r0p1
2253509	Programmer	Category C	A debug monitor step may cause an incorrect ICI value to be stacked in the RETPSR	r0p0	r0p1
2253502	Programmer	Category C	An interstating LDM may restart using an incorrect base address	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2279775	Programmer	Category C	An UNDEFINSTR fault could be prioritized over an INVSTATE fault for some invalid ICI values	r0p0	r0p1
2283939	Programmer	Category C	TCM requests that receive both a TGU fault and have forwarded data do not properly report a TGU fault	r0p0	r0p1
2467154	Programmer	Category C	S-AHB read bursts might terminate prematurely in case TCM Error is encountered by a speculatively prefetched read which immediately follows a read that encounters correctable ECC error	r0p0, r0p1	r0p2
2459389	Programmer	Category C	Executing VSCCLRM with an invalid ICI value may raise INVSTATE UsageFault	r0p0, r0p1, r0p2	r1p0
2456913	Programmer	Category C	Under limited circumstances, an LDM to a normal non-cacheable location followed by store to the same location might prevent the core from transitioning to lower power mode	r0p0, r0p1	r0p2
2388605	Programmer	Category C	A cacheable load which misses the data cache due to a correctable tag ECC error can return stale data if the cacheline it is reading is dirty	r0p0, r0p1	r0p2
2323280	Programmer	Category C	Word crossing store accesses that access the last word of the PAHB memory region might cause data corruption or incorrect fault behavior	r0p0	r0p1
2321988	Programmer	Category C	LSERR v/s NOCP Fault prioritization during PushStack	r0p0	r0p1
2319196	Programmer	Category C	DWT_PCSR can present next non-committed instruction when no instruction retirement occurs in the core	r0p0	r0p1
2287723	Programmer	Category C	The vector register file may not be cleared when tail chaining from a secure to non-secure exception	r0p0	r0p1
2312664	Programmer	Category C	Under limited circumstances, LDRD or VLDR instructions that access both device and non-device memory can deadlock the memory system	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2297046	Programmer	Category C	RTL incorrectly halts as it incorrectly computes vector catch on lockup entry	r0p0	r0p1
2335473	Programmer	Category C	A load or store multiple might cause an incorrect ICI value to be stacked in the RETPSR	r0p0	r0p1
2374615	Programmer	Category C	FP exception flags in the FPSCR might be set incorrectly after FP context creation following a security state change	r0p0	r0p1
2374006	Programmer	Category C	Undefined 16bit opcodes could raise a NOCP UsageFault instead of an UNDEFINSTR UsageFault	r0p0	r0p1
2374269	Programmer	Category C	Unprivileged debugger access allowed to read or write the DPDLPSTATE register	r0p0	r0p1
2371473	Programmer	Category C	The execution priority might be incorrect for a cycle when an IRQ's priority is updated in its handler	r0p0	r0p1
2397109	Programmer	Category C	An Instruction might be incorrectly sent to Embedded Trace Macrocell as partially completed instruction	r0p0, r0p1	r0p2
2311453	Programmer	Category C	Flaw in memory system power down may result in data corruption	r0p0	r0p1
2282154	Programmer	Category C	Under limited circumstances, a load to a normal non-cacheable location might not be properly ordered by a barrier instruction	r0p0	r0p1
2272772	Programmer	Category C	Debug read to FPSCR may return wrong default value	r0p0	r0p1
2276413	Programmer	Category C	Debug writes to privleged only bits of current state using DCRSR may not occur if UDE is enabled	r0p0	r0p1
2365142	Programmer	Category C	Direct cache access to Instruction cache data RAM cannot read half of the RAM locations	r0p0	r0p1
2355269	Programmer	Category C	A DWORD-sized store access to ITM_CIDR2 and ITM_CIDR3 can fault incorrectly on the access to the ITM_CIDR3 when the store is not privileged	r0p0	r0p1

ID	Area	Category	Summary	Found in versions	Fixed in version
2374351	Programmer	Category C	Trace fails to update PC on Imprecise bus fault under IESB fault escalation	r0p0, r0p1	r0p2
2631240	Programmer	Category C	Unexpected writes to DEBRs can potentially stop the outstanding linefills from completion	r0p0, r0p1, r0p2, r1p0, r1p1	Open
2601889	Programmer	Category C	Under limited circumstances, an ECC error on data cache RAMs can lead to data corruption	r0p0, r0p1	r0p2
2593278	Programmer	Category C	Under limited conditions, an imprecise abort may not be reported when a tag RAM ECC error is detected and data loss has occurred	r0p0, r0p1	r0p2
2589246	Programmer	Category C	An incorrect fault address is reported for an unprivileged aligned doubleword store to the STIR and RFSR registers when CCR.USERSETMPEND is set	r0p0, r0p1, r0p2, r1p0, r1p1	Open
2902253	Programmer	Category C	ITM can generate synchronization and timestamp packets when NIDEN=0	r0p0, r0p1, r0p2, r1p0, r1p1	Open
2666660	Programmer	Category C	An unusual combination of events can cause the core to hang instead of lock up on SOE entry	r0p0, r0p1, r0p2	r1p0
2674410	Programmer	Category C	An AXI load reading some portion of a word that an older store is writing might fail to report a BusFault in certain situations	r0p0, r0p1, r0p2	r1p0
2666664	Programmer	Category C	Under limited conditions, an unaligned MVE load reading the TCMs may return corrupted data if it encounters a transient BusFault on its TCM read	r0p0, r0p1, r0p2	r1p0
2746551	Programmer	Category C	Store address to DWT incorrect for certain VSTR operations	r0p0, r0p1, r0p2	r1p0
2757159	Programmer	Category C	Under limited circumstances, store operations do not update the data cache correctly	r0p0, r0p1, r0p2	r1p0
2682779	Programmer	Category C	After deactivating the instruction cache, self-modified code might not be executed correctly	r0p0, r0p1, r0p2	r1p0
2677972	Programmer	Category C	CTITRIGIN not connected to DWT_CMPMATCH for ETM = 0 configuration	r0p0, r0p1, r0p2	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
2705117	Programmer	Category C	L1D_CACHE_REFILL and L1D_CACHE_MISS_RD PMU events might be inaccurate	r0p0, r0p1, r0p2	r1p0
2614056	Programmer	Category C	Under limited conditions, a doubleword-aligned doubleword ITCM load can read data corrupted by an ECC error	r0p0, r0p1	r0p2
2640876	Programmer	Category C	Load data value for certain unaligned vector loads incorrect to DWT for data value matching	r0p0, r0p1, r0p2	r1p0
2651727	Programmer	Category C	Unprivileged debug transactions to the STIR and EVENTSPR registers can pend interrupts	r0p0, r0p1, r0p2	r1p0
2446484	Programmer	Category C	Simultaneous ECC errors on a store tag lookup and a load data read in the data cache can cause the ECC error for the load to not get corrected	r0p0, r0p1	r0p2
2439477	Programmer	Category C	A Secure read of ICSR_NS.VECTPENDING might return the Secure value	r0p0, r0p1	r0p2
2445488	Programmer	Category C	ETM incorrectly reports there are also additional serious exceptions when reporting entry into a serious exception	r0p0, r0p1	r0p2
2444277	Programmer	Category C	Store data value for certain MVE stores is incorrect to DWT for data value matching	r0p0, r0p1, r0p2	r1p0
2490920	Programmer	Category C	A malformed tail-predicated loop at the start of a Secure function may use the value of LTPSIZE without masking	r0p0, r0p1	r0p2
2452728	Programmer	Category C	Before RAM power is up and completes cache auto invalidation sequence, i.e. D$ is accessible, a cacheable store data might not be observed by a load to the same address after D$ is accessible	r0p0, r0p1	r0p2
2451384	Programmer	Category C	CHAIN PMU event unconnected to PMU EVENTBUS	r0p0, r0p1	r0p2
3030982	Programmer	Category C	Store data value for certain aligned vector store incorrect to DWT for data value matching	r0p0, r0p1, r0p2, r1p0, r1p1	Open
2988396	Programmer	Category C	DWT data value packets can be generated while the match packet from same access overflows	r0p0, r0p1, r0p2, r1p0, r1p1	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3319763	Programmer	Category C	ACTLR_S bit[11] can not disable No Write-Allocate mode	r1p0	r1p1
3078268	Programmer	Category C	Incorrect multi-copy atomicity ordering between Non-cacheable loads with forwarded store data	r0p0, r0p1, r0p2, r1p0, r1p1	Open

Non-Confidential

Errata descriptions

Category A

2286936
Aliased branch prediction in an IT block near an exception entry or BLXNS may result in incorrect execution

Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1

Description

When the branch predictor makes a prediction for a non-branch instruction inside an *If-Then* (IT) block while exception entry or BLXNS is in progress, the IT state may be misapplied for the non-branch instruction and for the following instructions.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- Branch prediction is enabled for the current Security state
- An IT instruction exists within the first 8 instructions of either an *Interrupt Service Routine* (ISR) or a BLXNS targeted function
- Branch prediction produces a prediction on a non-branch instruction

Implications

If this erratum occurs, the execution of some of the instructions in the IT block may execute as if they are not in an IT block.

Workaround

No software workaround is provided.

2355499
Under limited circumstances, write-through store operations do not update the data cache correctly

Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0. Fixed in r0p1

Description

A Write-Through store operation to an address resident in the cache updates the cache, then later writes to external memory. However, before writing the external memory, the updated cache line is invalidated or replaced. A subsequent load/store/PLD operation to the same line misses the cache and updates the cache with old data from the external memory.

Configurations affected

This erratum affects all configurations that include data cache.

Conditions

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

- A Write-Through store operation writes the cache.
- The cache line written by the store becomes invalidated or replaced.
- A younger preload, or non-overlapping load/store of the same cache line misses the cache and fills the line into the cache before the write-through store completes.
- The Write-Through store writes to external memory without updating the new cache line again.

Implications

If this erratum occurs, the cache line might be corrupted and subsequent read of an address within this cache line might return stale data.

Workaround

There is no direct workaround for this erratum.

Where possible, Arm recommends that you use the MPU to change the attributes on any Write-Through memory to Write-Back memory. If this is not possible, it might be necessary to disable the cache for sections of code that access Write-Through memory.

2615656
Under certain conditions, an unaligned MVE load can return incorrect data for the lower word in its second tick if an older store is writing the same word

Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

When a store is writing some of the data in the lower word of the second tick of a younger unaligned *M-profile Vector Extension* (MVE) load, the unaligned MVE load may return incorrect data for that word. This only occurs when there is an ongoing linefill for the cacheline that the store and load are accessing, and the store completes its cache write before the linefill gets the stale doubleword that the store overwrote from main memory.

Configurations affected

This erratum affects all configurations that include the MVE.

Conditions

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

1. A doubleword store or a series of smaller stores is writing all bytes of an aligned doubleword in cacheline L
2. The store or series of stores misses the cache
3. If there is not a linefill for cacheline L that is already ongoing due to some other data cache request, the store or series of stores starts a linefill for cacheline L
4. An unaligned MVE load is executed and the address of its second tick is within the doubleword in cacheline L that is targeted by the older store(s)
5. The store or series of stores must complete its cache write before the linefill gets back the doubleword of data from main memory that is targeted by the store(s)

Implications

The unaligned MVE load returns stale data for its second tick's lower word. Program flow might be altered depending on how software uses the value read.

Workaround

There is no workaround.

2467111
Conditional LCTP may use out of date flag values for bypass or update of LTPSIZE

Status

Fault Type: Programmer Category A
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

A conditional *Loop Clear with Tail Predication* (LCTP) might not prevent tail-predication from being applied to the next instruction in program order.

Configurations affected

This erratum affects configurations with *M-Profile Vector Extension* (MVE).

Conditions

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

1. *If-Then* (IT)
2. LCTP, positioned such that it receives a condition code from the IT instruction, which is executed in parallel with either of:
   a. A younger predication compatible instruction OR
   b. An older conditional flag setting instruction.

Implication

If the flag (NZCV) for the LCTP instruction condition resolution is not available soon enough, the LTPSIZE variable might either be set to an incorrect value, or bypass an out of data value to a younger instruction.

Workaround

Please contact Arm if a workaround is required.

Category A (rare)

There are no errata in this category.

Category B

2247184
Automatic EWIC register save/restore sequence in some rare cases disables EWIC capability of recognizing wakeup events

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

If the *External Wakeup Interrupt Controller* (EWIC) register access across the *External Private Peripheral Bus* (EPPB) is significantly slower than core register accesses, it is possible to enter EWIC sleep without EWIC being active.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when:

- The MCU is configured to use EWIC for wakeup event management (WICCONTROL == 0b1101 that is Automatic EWIC save/restore sequence on powerdown/up)
- Core is configured to enter sleep state on Return from Exception (Set SCR.SLEEPDEEP == 1 and SCR.SLEEPONEXIT == 1)

and the following sequence of conditions is met:

1. Execute Return from Exception (RFE) instruction
   - Core initiates low-power state entry sequence including automatic-sequence to configure EWIC
2. A valid wakeup event happens before automatic sequence completed
   - Core initiates clearing of EWIC state
3. Execute WFI as the next instruction to reenter sleep state before EWIC state is cleared

Implications

If this erratum occurs, the core enters sleep state without EWIC being active. As a result, the core never exits sleep state that can be exploited as a Denial of Service mechanism.

Workaround

The erratum conditions are not expected to be met in software. If necessary, use software sequence to perform EWIC register save/restore sequence.

2257706
Under limited circumstances, store operations do not update the data cache correctly

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

A store operation to an address resident in the cache, writes to external memory without updating the data cache.

Configurations affected

This erratum affects all configurations that include data cache.

Conditions

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

1. Cacheable memory is attributed as No Write-Allocate or write streaming is enabled (Secure version of ACTLR.DISNWAMODE is set to 0)
2. Two non-overlapping stores to different double words (64-bit chunks) of the same cache line occur
3. A younger preload closely following the stores, fills the line into the cache before the stores complete
4. Store writes to external memory without updating the cache

Implications

If this erratum occurs, the cache line may be corrupted and subsequent read of an address within this cache line may return stale data.

Workaround

All the following conditions should be met to avoid this erratum:

- Cacheable memory is attributed as Write-Allocate
- MSCR[2] is set to 0
- Secure version of ACTLR[11] is set to 1

When the workaround is applied, there may be a performance impact to some use cases that stream cacheable data from external memory.

2266165
An ignored bus fault can cause an operation to not make forward progress

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

Ignored bus faults can cause a dual issued AXI load operation not to make forward progress.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- An access receives a bus fault when CCR.BFHFNMIGN is set to 1
- The load operation is performed in close succession to another load operation and the older receives a bus fault
- The younger operation requires an external bus request

Implications

If this erratum occurs, the dual issued access hangs.

Workaround

To avoid this erratum, do not use CCR.BFHFNIMIGN.

2485388
An ongoing DSB does not wait for the automatic invalidation of the data cache to complete before retiring

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

After a reset, the caches in the processor are automatically invalidated by hardware. As the processor *Technical Reference Manual* (TRM) states, a DSB instruction will wait for the automatic invalidation of the instruction and data caches to complete before it stops executing. DSB instructions are stalled correctly when the automatic invalidation of the instruction cache is active. However, the automatic invalidation of the data cache will not by itself cause a DSB instruction to be stalled.

Configurations affected

This erratum affects all configurations that include data cache. Configurations with a data cache but without an instruction cache have an increased chance of the erratum occurring.

Conditions

This erratum occurs when all the following conditions are met:

- All other memory operations older than the DSB have completed
- The automatic invalidation of the data cache is ongoing
- The automatic invalidation of the instruction cache is either complete or is not being performed

Implications

If this erratum occurs, subsequent loads or stores will access main memory for their data accesses instead of reading or writing the cache until the automatic invalidation completes. This behavior does not follow the guidance that is provided in the processor TRM that executing a DSB instruction will cause the processor to wait for the automatic invalidation operation to finish.

Workaround

If software relies on the DSB to wait for the automatic invalidation sequence to complete before retiring, then the erratum can be avoided by polling the CPWRDN bit (bit [17]) in the MSCR (the MSCR address is 0E00_1E00). Software should then execute a DSB only after MSCR.CPWRDN is LOW.

2338043
The fields IDC, IXC, UFC, OFC and IOC in the FPSCR might be corrupted after the execution of double precision floating point multiply and accumulate instruction

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

The fields IDC, IXC, UFC, OFC and IOC in the *Floating Point Status and Control Register _(FPSCR) might be corrupted after the execution of _double-precision* (DP) *floating-point* (FP) multiply and accumulate instruction.

Conditions

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

- A double-precision floating-point multiply and accumulate instruction that gets cancelled midway through its execution due to various microarchitectural conditions such as an external interrupt request or a memory system fault on a memory operation
- Another double-precision floating-point multiply and accumulate instruction.

Implications

The fields IDC, IXC, UFC, OFC and IOC in the FPSCR may be set to 1 when they should be 0 after the execution of a double-precision floating-point multiply and accumulate operation.

Workaround

Please contact Arm if a workaround is desired.

2312674
Under limited circumstances, PLDs to the same cache line address as pending line-fills cause duplicate cache allocation and thus data corruption

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

A *Preload Data* (PLD) that checks the cache hit or outstanding line fills hit could create a false miss. The PLD in turn started another linefill where the same line has been filled into the D$.

Configurations affected

This erratum affects all configurations that include data cache.

Conditions

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

- A STR instruction/prefetcher/flushed LDR starts to fill the line into cache.
- A PLD is targeting the same cacheline as the existing linefill.

Implications

If this erratum occurs, two copies of the line can be filled into the cache causing data corruption.

Workaround

To avoid this erratum, do not use PLD.

2309351
Incorrect values may be read from FP registers and VPR

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

Incorrect register values may be used to generate addresses for *M-profile Vector Extension* (MVE) scatter-gather instructions or when storing the VPR with a VSTR instructions in some circumstances. This can occur if the EPU is placed in a separate power domain, and the core is configured to power down the *Extension Processing Unit* (EPU).

Configurations affected

This erratum affects configurations which include MVE and have placed the EPU in a separate power domain.

Conditions

This erratum occurs when all the following conditions are met:

- The core is configured to place the EPU in retention
- Rare internal timing conditions occur
- A scatter-gather VLDR/VSTR or VSTR VPR is executed

Implications

If this erratum occurs, incorrect register values may be used to generate addresses for the VLDR and VSTR instructions or stored as the VPR value by a VSTR instruction.

Workaround

To avoid this erratum, do not use power modes which make use of EPU state retention.

2296335
DWT Watchpoints might not work reliably on Data address and Data value comparator matches

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

The *Data Watchpoint and Trace* (DWT) comparators can be set up to match on a Data Address or Data Value match, and based on the action programming can generate a debug event to the core. This erratum causes watchpoint hits to happen or be missed under certain conditions, resulting in unreliable debug event generation on Data Address and Data Value match programming in the DWT.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met:

- The DWT Comparator n is implemented, where n = 0-7.
- DWT_COMPn supports Data Address or Data Value matching and is programmed to match on Data Value or Data Address.
- DWT_COMPn is programmed to generate debug event action on a match.
- A comparator match on a Data Address or a Data Value that is programmed occurs near a execution flow change such as: a branch mispredict, a condition code failure, or an exception.

Implications

- A false watchpoint hit can be reported back to the core.
- A watchpoint hit can be missed.
- A false CMPMATCH can be triggered to ETM.

Workaround

- No workaround is available for sample silicon.

2288808
Under limited circumstances, loads do not properly hazard other cache updates which cause a duplicate entry to be placed in the data cache causing data corruption

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

A Non-cacheable load access that receives incomplete data forwarding may stop a closely executed Cacheable access to not observe a linefill into the cache. The load that failed to observe the linefill then requests an additional copy of the address that is already in the data cache. This results in a duplicate entry in the data cache.

Configurations affected

This erratum affects all configurations that include data cache.

Conditions

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

1. The load crosses into a Cacheable region or is dual issued with a Cacheable load and the access misses the cache
2. A Non-cacheable load is forwarded some but not all of its data from an older store and requires a bus request for the remaining data
3. The data is filled into the cache before the load requests it
4. The load incorrectly misses the cache and requests that a second copy to be filled into the cache

Implications

If this erratum occurs, two copies of the line can be filled into the data cache causing data corruption.

Workaround

To avoid this erratum, set the Secure version of ACTLR[17] to 1.

Under limited conditions this workaround might cause the processor to lose atomicity ordering of Normal memory with other agents where no synchronization is performed.

2295129
From an unprivileged debugger, cache maintenance and ERRDEVID registers are not accessible while ERRIIDR is accessible

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

Accesses to internal registers from an unprivileged debugger have register specific permissions. Due to this erratum, certain data and instruction cache maintenance registers and the Error Record Device ID Register, ERRDEVID are not accessible from an unprivileged debugger and will return a fault instead of the access being permitted. Accesses to the ERRIIDR from an unprivileged debugger will not fault when the access is expected to fault.

Configurations affected

This erratum affects all configurations of the Cortex-M85 processor configured with Unprivileged Halting debug.

Conditions

This erratum occurs when DAUTHCTRL_S.UIDAPEN = 1 or DAUTHRCTRL_NS.UIDAPEN =1 and

For the erratum part concerning accesses that should have permissions but receive an error, either of the following two conditions apply:

- There is a write access through an unprivileged *Debug Access Port* (DAP) request to one of the following registers
```
0xE000EF70, DCCIMVAC
0xE000EF74, DCCISW
0xE000EF68, DCCMVAC
0xE000EF64, DCCMVAU
0xE000EF6C, DCCSW
0xE000EF50, ICIALLU
0xE000EF58, ICIMVAU
```

- Or a read access is made to the following register
```
0xE0005FC8, ERRDEVID
```

For the erratum part where an access is permitted when it is expected to fault: the following condition applies

- There is a read access through an unprivileged DAP request to:
```
0xE0005E10, ERRIIDR
```

Implications

If a debugger inserts an instruction such as a BKPT operation into memory, cache maintenance operations are required to guarantee coherency between the data cache and the instruction cache, such that the new instruction is executed rather than a 'stale' instruction which was previously fetched into the instruction cache. Due to this erratum:

- The cache maintenance operations cannot be applied if the debugger is unprivileged.
- The ERRDEVID register cannot be read by an unprivileged debugger to determine the number of Error records supported by the Armv8.1-M *Reliability, Availability, and Serviceability* (RAS) Extension.
- The read-only register, ERRIIDR, can be read by an unprivileged debugger when it should fault.

Workaround

There is no workaround for this erratum.

2288773
A VLDRD executing in big-endian mode returns incorrect data if the load reads a word from device memory that is fully predicated when the other word isn't fully predicated

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

A big-endian vector load with doubleword element size that has a predicted word accessing device memory may incorrectly swap its words when updating the destination register. This results in the predicated data from memory being returned to the non-predicated word and no update to the non-predicated word.

Configurations affected

This erratum affects all configurations that include the Armv8.1-M Vector Extension.

Conditions

All of the following conditions must be met for this erratum to occur:

- CFGBIGEND is set when the VLDRD begins executing
- The predicate for the VLDRD fully predicates only one of the words it is to read
- The word that is fully predicated must be in device memory
- The access is to a TCM, AXI, or PAHB target

Implications

The VLDRD returns incorrect data. Program flow might be altered depending on how software uses the value read.

Workaround

Do not use big-endian VLDRD instructions accessing device memory.

2315530
IESB derived fault hang during entry

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

The core might hang during exception entry if an escalated imprecise fault is encountered while FPCA is in a state not consistent with other controls.

Configurations affected

This erratum affects configurations with MVE or FPU.

Conditions

This erratum occurs when all the following conditions are met:

- AIRCR.IESB== 1
- (FPCCR.LSPACT== 1 OR FPCCR.LSPEN==1 OR (non secure state and nsacr_i.cp10 == 0))
- CONTROL.FPCA== 1
- CP enabled in at least one of CPACR_S, CPACR_NS

Implications

If IESB is enabled, the core might hang if there is an asynchronous bus fault before entry is taken. For r0p0 release, most RAS features were defeatured, so the only sources of these faults are bus errors on stores. Therefore the utility of setting IESB on sample parts is low, and if IESB is set, the risk of encountering this errata is also low. That this only effects sample silicon at LAC quality was factored into the severity.

Workaround

If untrusted code is going to be run, disable AIRCR.IESB. If code can be trusted not to corrupt FPCA or LSPACT, then disable Lazy stacking by clearing FPCCR.LSPEN.

2425308
When an MVE store encounters a fault, in some cases the FAR is not updated to the address of the beat that reports the fault

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

On a fault, *M-profile Vector Extension* (MVE) stores always update the *Fault Address Register* (FAR) to the doubleword-aligned base address of the current tick, even when the lowest beat is predicated out or ECI indicates that the lowest beat has already been completed.

Configurations affected

This erratum affects all configurations that include the Armv8.1-M Vector Extension.

Conditions

An MVE store is only affected by this erratum if it reads more than a single word. The MVE store will update the FAR address incorrectly to the doubleword-aligned base address of the current tick instead of the word-aligned address of the upper beat of the tick in one of three scenarios:

1. Only the upper beat in the tick gets a fault and it is not predicated out.
2. Both beats in the tick get faults. The lowest beat is predicated out or ECI indicates it has already been completed. The upper beat is not predicated out.
3. Both beats in the tick get faults. Neither beat is predicated out nor has the lowest beat already completed. The lowest beat's fault is an ignored BusFault.

Implications

The MMFAR, BFAR, or SFAR (depending on the type of fault) is updated to the address one word below where the actual fault occurred. The new FAR address is guaranteed to be within the same tick as the correct fault address.

Workaround

We can work around this erratum if software handles the uncertainty about what the correct fault address is. Since the correct fault address is either what is reported or the address one word above that, software can manually inspect each address with a smaller instruction (that is, a store for each beat that writes the same data that the MVE store would write to that beat). A fault on a smaller access will indicate which fault address is correct. If the fault was transient and neither smaller store gets a fault, then we can continue on since we have now written the data correctly.

2371188
A flag writing external coprocessor instruction may cause incorrect bypass from younger conditional instructions

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

A condition false flag updating MRC or MRC2 instruction may result in incorrect bypassing between a pair of younger instructions, causing data to be bypassed from a conditional false instruction if a rare combination of micro-architectural events occurs.

Configurations affected

This erratum affects all configurations.
The instructions that trigger this erratum are not expected if external coprocessors are not implemented.

Conditions

This erratum might occur when the following sequence of instructions occurs:

1. An IT instruction.
2. An APSR writing MRC (Rt=15) which resolves as condition false from coprocessor 0-7 (external coprocessors).
3. Within the next 5 instructions, a condition false instruction writing a general-purpose register followed by a condition true instruction reading that same register.
4. Specific micro-architectural timing conditions.

Implications

This erratum might result in data-corruption, as data might bypass between instructions that should not occur. This erratum does not impact FP registers.

Workaround

Note that this erratum does not affect designs which do not use external coprocessor.

2627033
An LDR following a flushed LDM targeting the same Non-cacheable line can return corrupted data

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

An LDM from Non-cacheable region and its starting address is not doubleword-aligned. When the LDM is flushed, a following LDR that targets the same doubleword but the address is not part of the LDM will not generate bus request. This causes LDR to get UNKNOWN data from the M-AXI interface.

Configurations affected

This erratum affects all configurations that do not include data cache.

Conditions

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

1. An LDM from Non-cacheable memory and the starting address of the LDM is not doubleword-aligned
2. The LDM is flushed due to external interrupts or bus abort
3. No load operation to the AXI memory region that is outside of the 32-byte granule accessed by LDM
4. An LDR from the address that is below the LDM starting address but in the same doubleword

Implications

The load might get corrupted data.

Workaround

To avoid this erratum, set the ACTLR_S[16] to 1. However, this will impact the performance of Non-cacheable operations.

2722045
Under certain conditions, a load or store crossing into an MPU UNDEFINED region can report an unaligned UsageFault instead of the expected MemManage Fault

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2. Fixed in r1p0

Description

A load or store which attempts to access an MPU UNDEFINED region should normally report a MemManage Fault on its access. However, if the load or store is unaligned and is crossing into the *Memory Protection Unit* (MPU) UNDEFINED region from below, it might report an unaligned UsageFault instead of the expected MemManage Fault.

Configurations affected

This erratum affects all configurations.

Conditions

All the following conditions must be met for this erratum to take effect:
1. The load or store must be unaligned.
2. The load or store must be crossing into an MPU UNDEFINED region from below (i.e. the lowest byte(s) of the access must be outside the UNDEFINED region and some higher byte(s) must be within the UNDEFINED region).
3. The default system address map for the MPU UNDEFINED region must be of device type, but the default system address map is not selected.
4. The load or store cannot be crossing into or out of PPB memory.
5. The load or store cannot be a kind of instruction that always gets an unaligned UsageFault if it is unaligned (e.g. exclusives, scatter/gather ops, load acquires, and store releases always fault if they are unaligned).
6. The CCR.UNALIGN_TRP register bit cannot be set.

Implications

An unaligned UsageFault is reported for the load or store when a MemManage Fault is architecturally expected instead.

Workaround

The erratum can be worked around by adhering to some restrictions for MPU region mapping:

1. If possible, a MPU region mapped to normal memory should never be just below an UNDEFINED MPU region with a default device memory mapping.
2. If a MPU region mapped to normal memory must be just below an UNDEFINED MPU region whose default mapping is device, a new guard MPU region should be added at the boundary between the normal and UNDEFINED regions with two properties:
a. The guard region must be non-device.
b. The guard region must have no access permissions.

Date of issue: April 16, 2024
Cortex-M85 AT640 and Cortex-M85 with FPU AT641
Software Developer Errata Notice
Version: 14.0

2449355
A word-crossing store which gets an ignored BusFault for one of its words can cause a hang when dual-issued with a younger Device load

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

When a word-crossing store gets an ignored BusFault on only one of the words it is writing, the ignored BusFault may cause a hang if the store is dual-issued with a younger Device load.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- The store must cross a word boundary.
- The store must be dual-issued with a younger load.
- The load must have at least one lane with Device memory attributes. This Device lane of the load cannot be predicated out.
- The store must get an ignored BusFault for one of the words it is writing to. The other word cannot have any fault and cannot be predicated out.
- The word the store is writing to that gets an ignored BusFault must be in either *Tightly Coupled Memory* (TCM) or *Internal Private Peripheral Bus* (IPPB) memory region.

Implications

If this erratum occurs, the memory system hangs and deadlock can only be broken through an interrupt.

Workaround

To avoid this erratum, do not ignore BusFaults.

3190818
Under limited circumstances, LDM to normal non-cacheable AXI location can not complete

Status

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

Description

In specific microarchitectural conditions and AXI configurations, *Load Multiple* (LDM) to non-cacheable location might fail to make forward progress.

Configurations affected

This erratum affects all configurations with data cache.

Conditions

This erratum occurs when either one of the two sets of conditions are met:

1. Conditions set A
   - *Cache Maintenance Operation* (CMO) to data cache
   - Several store instructions, in program order either before or after the CMO, to AXI memory space with mixed memory types, some non-cacheable, some cacheable
   - LDM to non-cacheable location in program order after the CMO and sores, the accessed memory address overlapping with one of the non-cacheable stores in the store buffer
   - An interrupt or asynchronous exception is being invoked
2. Conditions set B
   - Store initiates a write transaction on AXI and does not receive WREADY
   - At least two line fill requests initiate read transactions on AXI
   - Core de-asserts RREADY due to internal u-arch conditions
   - RREADY and WREADY both de-asserted for over extended time
   - RREADY is re-asserted to complete the linefill
   - LDM to normal non-cacheable location initiates read transactions on AXI

Implications

Core fails to make forward progress.

Workaround

This erratum can be avoided by setting the ACTLR_S[16] to 1. However, this workaround will impact the performance of Non-cacheable LDM operations.

The Cortex-M85 Technical Reference Manual documents ACTLR_S[16] as reserved. This workaround replaces ACTLR_S[16] access restrictions, as set out in the Technical Reference Manual, and enables read-write access to ACTLR_S[16].

2958637
Certain PMU EVENTBUS events not driven correctly on M85

Status

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

Description

Cortex-M85 supports an EVENTBUS which exports *Power Management Unit* (PMU) events as a single cycle bus on the external interface. Due to this erratum certain events listed below are unavailable on the bus.

List of events: MVE_STALL_RESOURCE, CTI_TRIGOUT4, CTI_TRIGOUT5, CTI_TRIGOUT6, CTI_TRIGOUT7, TRCEXTOUT0, TRCEXTOUT1, and CHAIN.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

This erratum occurs when the following conditions are met:

- The listed event is enabled by programming the appropriate control register and the EVENTBUS is enabled.

Implications

The listed signals on the EVENTBUS will always be 0. The STL library needs to be rebuilt to use the formula "MVE_STALL_RESOURCE_MEM | MVE_STALL_RESOURCE_FP | MVE_STALL_RESOURCE_INT" instead of the disconnected MVE_STALL_RESOURCE signal on the EVENTBUS.

Workaround

These events can be counted by programming the PMU accordingly. The MVE_STALL_RESOURCE can be determined by observing the MVE_STALL_RESOURCE_MEM, MVE_STALL_RESOURCE_FP, and MVE_STALL_RESOURCE_INT on the EVENTBUS.

3175626
AXI hang due to dependency between read data channel and write response channel

Status

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

Description

In specific AXI interconnect configurations and microarchitectural conditions, load can not make forward progress due to WAW hazard between eviction and stores.

Configurations affected

This erratum affects all configurations with data cache.

Conditions

This erratum occurs when all the following conditions are met:

- Series of non-allocating stores initiate write transaction on AXI, waiting for a buffered write response (BRESP)
- Two or more data cache line fill requests are triggered by load/store/prefetcher initiating read transactions on AXI
- The line fill requests trigger cache line evictions to accommodate the impending line fills.
- The evictions are serialized behind the write transactions due to WAW hazards (not exact address match)
- Valid buffered write response (BVALID) for the pending write transactions are not provided from the interconnect until valid read responses (RVALID) are accepted for the read transactions
- The line fills wait for eviction to complete, preventing acceptance of further read responses (RVALID)

Implications

The load awaiting a line fill prevents all younger instructions from completing.

Workaround

For system with interconnects that serialize between AXI read data and write responses, the system can work around the erratum by attributing all cacheable memory as write-through. When the workaround is applied, there may be a performance impact to some use cases.

Category B (rare)

2434209
Certain stall cases can cause an unaligned MVE load to incorrectly zero some of its return data or fail to report a fault

Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0, r0p1. Fixed in r0p2

Description

In three different conditions, a stall (combined with rare micro-architectural conditions) may cause an unaligned *M-profile Vector Extension* (MVE) load to zero out the data in the upper beat of its first tick even when that data is not predicated out. The stall may also cause a fault in either beat of the first tick to not be reported when the tick completes.

Configurations affected

This erratum affects all configurations that include the MVE.

Configurations with the MVE that also include the *Embedded Trace Macrocell* (ETM) or the *Instruction Trace Macrocell* (ITM) have an additional condition that the erratum can occur.

Conditions

This erratum occurs when an unaligned MVE load is being executed and one of the following three conditions applies. All three conditions cause a stall.

- The unaligned MVE load gets an ignored BusFault.
- The unaligned MVE load crosses into the *Private Peripheral Bus* (PPB) memory space (crossing out of the PPB memory space does not have the same effect).
- The trace stall is asserted. The trace stall can only be asserted if either the ETM or ITM is included and at least one of the following is true:
  - The ETM is included. Invasive debug is enabled. TRCSTALLCTLR.ISTALL is set.
  - The ITM is included. DEMCR.TRCENA is set and ITM_TC.STALLENA is set.

Implications

If this erratum occurs, zeros may be returned instead of the correct data for the upper beat of the first tick of an unaligned MVE load. Alternatively, a fault on either tick of the first beat of the unaligned MVE load may fail to get reported.

Workaround

- To avoid the first condition of this erratum (The unaligned MVE load gets an ignored BusFault): Do not ignore BusFaults.
- To avoid the second condition of this erratum: Do not use unaligned MVE loads targeting PPB memory for only one lane. Such loads are not expected in typical software.
- To avoid the third condition of this erratum (ETM or ITM stalls): Disable the trace stalls.

2705514
Under limited circumstances, an LDM following a flushed LDM from Non-cacheable region can result in deadlock

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2. Fixed in r1p0

Description

An LDM from Non-cacheable region and its starting address is not the first two word addresses in a 32-byte line. When the LDM is flushed, a following LDM partially overlaps with the flushed LDM. The LDM might fail to make forward progress.

Configurations affected

This erratum affects all configurations.

Conditions

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

1. An LDM/POP/VLDM/VPOP from Non-cacheable memory and the starting address of the instruction is not the first two word addresses in the 32-byte line
2. Specific rare micro-architectural conditions occur
3. The LDM/POP/VLDM/VPOP is flushed due to external interrupts, imprecise abort or precise bus abort
4. Typically the interrupted LDM will continue after returning from handler, if it is not resumed or before it is resumed, no load from AXI out of the 32-byte line, A second LDM/POP/VLDM/VPOP whose starting word address does not belong to the flushed LDM, but the second word address overlaps with the first LDM.

Implications

LDM/POP/VLDM/VPOP will not request data from AXI which leads to hang and can only resume progress with a higher priority interrupt.

Workaround

Given the rareness of the conditions to hit the bug, no workaround should be required. If a workaround is wanted to avoid this erratum, set the ACTLR_S[16] to 1. However, this will impact the performance of Non-cacheable LDM operations.

Category C

2242544
Data corruption may be observed if there are consecutive reads to different Wakeup Event Mask registers

Status

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

Description

Read to a Wakeup Event Mask register may cause data corruption on read to a different Wakeup Event Mask register that immediately follows.

Configurations affected

This erratum affects all configurations.

Conditions

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

- A read to an EVENTMASKA or EVENTMASKn register occurs
- A read to a different EVENTMASKA or EVENTMASKn register immediately follows the first read

Implications

If this erratum occurs, consecutive reads to different Wakeup Event Mask registers might result in data corruption on the second read.

Workaround

On sleep entry, the Wakeup Event Mask registers are expected to be used as part of processor state transfer to the *External Wakeup Interrupt Controller* (EWIC). This can be done in two ways, both of which are not affected by this erratum:

- Automatic hardware save mechanism that transfers the required state from the processor to the EWIC. This mechanism does not use back to back reads to EVENTMASKA or EVENTMASKn registers.
- Software performs the required state transfer to EWIC by reading and writing the relevant registers.

Software will typically write the EWIC_MASKA or EWIC_MASKn register following a read to respective EVENTMASKA or EVENTMASKn register.

No workaround is needed if the Wakeup Event Mask register reads are only performed as in the expected use cases above. However, if a read of EVENTMASKA or EVENTMASKn register is done outside those scenarios, it should be followed by another read that is not accessing any *Internal Private Peripheral Bus* (IPPB) registers.

2434651
An aligned doubleword MVE load reading the ITCM with one word predicated out might read data corrupted by an ECC error for the non-predicated word

Status

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

Description

When an aligned doubleword *M-profile Vector Extension* (MVE) load reading the *Instruction Tightly Coupled Memory* (ITCM) has one of its words fully predicated out, an *Error Correcting Code* (ECC) error in the memory read for the other word that is not predicated out might go undetected. The data read for that word is not corrected and the ECC error is not reported.

Configurations affected

This erratum affects configurations that include all the following:

- Armv8.1-M Vector Extension
- ECC protection for the TCMs
- Nonzero TCM wait-states for at least the ITCM

Conditions

This erratum occurs when all the following conditions are met:

- The load must be an aligned MVE
- The load must be reading a doubleword from the ITCM. The erratum will not occur if the load is reading one word from the ITCM and another word from a different memory
- The load must be doubleword-aligned
- The load must have one word predicated out
- ITCMWAIT must be asserted for at least one cycle after the load requests data from the ITCM

Implications

The load returns data corrupted by an ECC error. The ECC error, which can be correctable or fatal, is not reported to the RAS registers or on the DME bus.

Workaround

The erratum can be avoided if aligned MVE loads which read a doubleword are not used to read the ITCM.

2434207
An MVE tick that encounters faults in both beats may report the FAR address incorrectly if the fault types for each beat are different

Status

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

Description

When the lower beat of an *M-profile Vector Extension* (MVE) tick gets an ignored BusFault and the upper beat gets a non-ignored fault (either a non-ignored BusFault or non-BusFault), the *Fault Address Register* (FAR) may be updated for the non-ignored fault using the address for the ignored BusFault.

Configurations affected

This erratum affects all configurations that include the MVE.

Conditions

This erratum occurs when all the following conditions are met:

- A load or store MVE instruction is being executed.
- The MVE gets an ignored BusFault in the lower beat in a tick.
- The MVE gets a non-BusFault or non-ignored BusFault in the upper beat in the same tick.
- Neither beat can be predicated out.

Implications

If this erratum occurs, the *MemManage Fault Address Register* (MMFAR), *BusFault Address Register* (BFAR), or *Secure Fault Address Register* (SFAR) (depending on the type of fault in the upper beat of the tick) is updated with the fault address for the lower beat in the MVE tick when the upper beat's address should be reported.

If the non-ignored fault in the upper beat of the tick is a BusFault, the reported BusFault type may also be incorrect. The BusFault type is only incorrect if the lower and upper beat have different kinds of BusFault.

Workaround

There is no workaround. However, MVE instructions are not expected to be used when BusFaults are ignored.

2432946
ETM reports wrong address on tail-chain during sleep on exit

Status

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

Description

Incorrect address traced on *Embedded Trace Macrocell* (ETM) when a tail chain occurs during sleep on exit.

Configurations affected

This erratum affects configurations with ETM.

Conditions

This erratum can occur when the following conditions are met:

- scr.sleeponexit = 1 for the current handler mode
- ETM trace is enabled for the current handler mode

And then the following events occur in order:

1. The current handler exits to thread mode, triggering an entry to sleep instead of completing the return.
2. An interrupt occurs, causing a tail-chain to the new handler.

Implications

The address traced for the tail chain should be 0xFFFFFFFE, but instead will be the preferred return address of the prior exception taken. The E1E0 field of the packet will be correct however, which should prevent incorrect trace decompression.

Workaround

Because trace decompression is not impacted, no workaround is needed.

2266957
Incorrect execution priority may be applied for a short duration in rare cases after the start of an IRQ return

Status

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

Description

A small window exists during which an incorrect execution priority can be applied by the *Nested Vector Interrupt Controller* (NVIC) to some synchronous faults.

Configurations affected

This erratum affects all configurations in r0p0.

This erratum affects if NUMIRQ > 464 in r0p1.

Conditions

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

1. A Non-secure IRQ is activated
2. A Secure subroutine is called
3. The IRQ exception return begins from within the Secure subroutine while EXCRET.ES is zero
4. INVER, INVPC, LSERR, or NOCP is generated during exception return sequence

Implications

A synchronous INVER, INVPC, LSERR, or NOCP fault that occurs in the two-cycle window when the incorrect execution priority is applied might be escalated to HardFault even if it has enough priority to preempt.

Workaround

No workaround is required for this erratum.

2265059
A store with memory attributes that do not match a younger LDM to the same address may cause a hang

Status

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

Description

A cacheable store cannot drain when there is a younger LDM from the same cache line with memory attributes has been remapped to Non-cacheable without proper barriers.

Configurations affected

This erratum affects all configurations include data cache.

Conditions

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

1. Execute store to a cacheable address A
2. Program *Memory Protection Unit* (MPU) to change memory attribute of A to Non-cacheable
3. Execute LDM which includes address A as part of the access

Implications

If this erratum occurs, the store may not complete and therefore, hangs the core.

Workaround

To avoid this erratum, execute a DSB after modifying the MPU configuration.

2253509
A debug monitor step may cause an incorrect ICI value to be stacked in the RETPSR

Status

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

Description

A debug monitor stepping event may cause a zero stacked in the ICI field of the RETPSR when a valid non-zero ICI value would be expected.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- Debug monitor stepping is active
- The instruction following the next instruction to execute is a valid *Interrupt Continuable Instruction* (ICI) such as LDM, STM, VLDM, or VSTM
- A non-zero ICI value is present in EPSR on completion of the next instruction

Implications

When these conditions occur during a debug monitor stepping event, the stacked RETPSR ICI value may contain zero instead of the valid non-zero value. If this erratum occurs, the only effect would be a restart instead of resume of the affected load or store multiple. No other register or memory corruption will occur.

Workaround

No workaround is required for this erratum.

2253502
An interstating LDM may restart using an incorrect base address

Status

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

Description

A resumed LDM causing an interstating function return may not properly restart after encountering a
synchronous fault during unstacking of the FNC_RETURN stack frame or during the integrity checks of
the unstacked IPSR value.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- A non-faulting LDM with writeback which loads the PC with a value of FNC_RETURN.
- The LDM with an encoding that expects the base register to contain its final value when interrupted
  such as non-SP as base, or LDMDB with SP as base
- A valid non-zero ICI value that causes the LDM to resume
- A faulting condition during unstacking of the partial RETPSR and ReturnAddress or during integrity
  checks of the unstacked IPSR value.

Implications

If this erratum occurs, the base register value may not be properly restored. This may cause an incorrect
base value to be used if the LDM is restarted later. This erratum does not affect instructions that are
typically used for function returns (POP).

Workaround

No workaround is required. Erratum conditions are not expected to be met in actual software.

2279775
An UNDEFINSTR fault could be prioritized over an INVSTATE fault for some invalid ICI values

Status

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

Description

INVSTATE faults are raised by the core for load and store multiple instructions which are not resumable or execute with ICI values that do not match the register list of the instruction. If those instructions also encounter a condition which would raise an UNDEFINSTR fault, the UNDEFINSTR fault could be prioritized over the INVSTATE fault.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- A load or store multiple instruction executes with an ICI resume value for a register that is not in the list, first in the list, or is not resumable
- The instruction also encounters a condition which raises an UNDEFINSTR fault

Implications

If this erratum occurs, the UNDEFINSTR flag instead of the INVSTATE flag, could be recorded in the UFSR, UsageFault Status Register when the UsageFault exception is taken.

Workaround

No workaround is required for this erratum.

2283939
TCM requests that receive both a TGU fault and have forwarded data do not properly report a TGU fault

Status

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

Description

Loads that fully forward data but receive a TGU fault may have their TGU fault masked.

Configurations Affected

This erratum affects all configurations include TCMs.

Conditions

This erratum occurs when all the following conditions are met:

- A load hazards and receives all of its data from older store to the same TCM location that has not yet written the TCM.
- The load has a TGU fault where the previous store to the location did not.
- The SAU and IDAU do not return a fault for this load.

Implications

A non-secure load to the TCM alias address may have its TGU fault masked by erroneously forwarding data from a secure load to the same address

Workaround

To workaround this erratum, ensure that the SAU or IDAU are configured to cause a fault for the load.

2467154
S-AHB read bursts might terminate prematurely in case TCM Error is encountered by a speculatively prefetched read which immediately follows a read that encounters correctable ECC error

Status

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

Description

When a read prefetched on behalf of a S-AHB read burst operation encounters a *Tightly Coupled Memory* (TCM) Error in the shadow of an ongoing read which has encountered a correctable *Error Correcting Code* (ECC) error, the burst will terminate prematurely. This will therefore prevent corrected data of the read to be sent out on the S-AHB interface.

Configurations affected

This erratum affects all configurations when ECC is enabled.

Conditions

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

1. ECC is enabled
2. S-AHB read burst operation is accepted by the processor to access the TCM
3. Processor internally generates subsequent read transactions of the burst so that read data can be returned on consecutive clock cycles on the S-AHB interface
4. TCM read to address A encounters a correctable ECC error
5. While the ECC error is being corrected, the subsequent generated read transaction to address B encounters TCMERR
6. Read burst operation is terminated before the data for address A is returned to the S-AHB interface

Implications

During a burst with a correctable error and a fatal error, the fatal error may be reported on the incorrect response.

Workaround

There is no workaround for this erratum.

2459389
Executing VSCCLRM with an invalid ICI value may raise INVSTATE UsageFault

Status

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

Description

The Cortex-M85 restarts execution of VSCCLRM instructions which are executed with a valid, non-zero ICI value. For implementations that restart execution for non-zero ICI values it is expected that VSCCLRM will also restart execution for invalid ICI values, but the Cortex-M85 will raise an INVSTATE UsageFault instead.

Configurations affected

This erratum affects configurations with floating-point support or *M-profile Vector Extension* (MVE) support.

Conditions

This erratum occurs when a VSCCLRM is executed with an ICI value indicating a resume register and that either of the following applies:

- The register is not in the list
- The register is the first register in the list of the VSCCLRM

Implications

If a VSCCLRM is executed with an invalid ICI resume value, the VSCCLRM will raise an INVSTATE UsageFault rather than restarting execution.

Workaround

No workaround is required. Erratum conditions are not expected to be met in typical software.

2456913
Under limited circumstances, an LDM to a normal non-cacheable location followed by store to the same location might prevent the core from transitioning to lower power mode

Status

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

Description

A low-power transition or warm reset request fails to clear the buffered load data and cannot drain the store targeting the same non-cacheable location as the buffered load.

Configurations affected

This erratum affects all configurations.

Conditions

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

1. Specific microarchitectural conditions occur on an LDM
2. No load operation to the AXI memory region that is outside of the 32-byte granule accessed by LDM
3. A store that falls to the same 32-byte granule as the LDM
4. A lower power transition or warm reset request

Implications

The core cannot transition to lower power mode or warm reset.

Workaround

There is one workaround for this erratum:

- The normal usage case would expect a DSB to drain the buffers before going to lower mode.

2388605
A cacheable load which misses the data cache due to a correctable tag ECC error can return stale data if the cacheline it is reading is dirty

Status

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

Description

When a cacheable load misses the data cache due to a correctable tag ECC error, the load can return stale data if the cacheline that it missed is dirty. The load can return stale data only if all the bytes are dirty in at least one of the four doublewords within the cacheline and the load is reading at least one byte from that dirty doubleword.

Configurations affected

This erratum affects all configurations that include the data cache and ECC protection for the data cache.

Conditions

- The location of interest is cacheable.
- A DW-sized store or a series of smaller stores updates all bytes of a doubleword and misses the cache.
- The linefill has not completed filling the cache before the store updates the cache.
- A load of any bytes from the DW updated by the store misses the cache due to a correctable tag ECC error.

Implications

In the presence of correctable TAG ram ECC errors, most of the time the errors are detected and corrected. However, there are a few cases in which the errors are not handled correctly, which can result in incorrect data being returned for a load.

Workaround

There are no workarounds for this erratum.

2323280
Word crossing store accesses that access the last word of the PAHB memory region might cause data corruption or incorrect fault behavior

Status

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

Description

Store accesses that cross from the penultimate word to the last word of PAHB memory that do not generate an alignment fault might cause data corruption or incorrect fault behavior.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- An element aligned store that accesses both the penultimate and last word of PAHB space occurs.
- The operation is big-endian or an MVE access.

Implication

- The last word of the PAHB region is incorrectly identified as a PPB, which causes access and accesses to be performed as little-endian.
- The operation will be performed as little endian and will fault if the operation type to PPB space would cause a fault.
- This is not an expected use case for store accesses to PAHB memory.

Workaround

Do not perform store operations that cross a word boundary into the last word of the PAHB region.

2321988
LSERR v/s NOCP Fault prioritization during PushStack

Status

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

Description

During exception entry stacking, if FP register stacking would encounter both a LSERR and a NOCP fault, the NOCP fault will take priority, but the LSERR should take priority during implicit stacking.

Configurations affected

This erratum affects all configurations with MVE!=0 or FP!=0

Conditions

This erratum occurs when all the following conditions are met:

- CONTROL.FPCA==1
- (FPCCR_S.S ? FPCCR_S.LSPACT : FPCCR_NS.LSPACT)==1
- CPACR or NSACR forbids access to the FP register file for the current security and privileged level.
- An exception is triggered.

Implications

The NOCP fault might be triggered when the LSERR fault should have been trigged. The NOCP fault is higher priority than the LSERR fault for normal FP register accessing instructions, but the fault priority should place LSERR as higher priority during automatic stacking during exception entry. This means it is possible for the usage-fault handler to be invoked when the specification calls for the secure-fault handler to be invoked. For some software configurations, this might allow an unsuccessful attempt to evade security by incorrectly setting FPCA to go undetected, as the usage fault might be directed to the non-secure usage fault handler.

Workaround

To avoid the consequences of this errata, the following steps can be taken:

- In the non-secure usage-fault handler, check the value of FPCCR.LSPACT and CONTROL.FPCA to detect attempts to misconfigure the floating point context.
- Avoid using lazy-stacking for secure mode by setting FPCCR_S.LSPEN to 0.

2319196
DWT_PCSR can present next non-committed instruction when no instruction retirement occurs in the core

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1

Description

Program Counter (PC) sampling is an optional component provided through DWT_PCSR. When DWT_PCSR returns a value other than 0xFFFFFFFF, the returned value is an instruction that has been committed for execution. A read of DWT_PCSR does not return the address of an instruction that has been fetched but not committed for execution. When no instruction retires in the core, the next reported instruction via DWT_PCSR will be based on the instruction PCs received from the core rather than the last retired instruction.

Configurations affected

This erratum affects all configurations of the processor configured with Halting debug.

Conditions

This erratum occurs when all the following conditions are met:

- The DWT PCSR is implemented
- DWT_PCSR is read in a cycle where no instructions are retired

Implications

If this erratum occurs, a DWT_PCSR will provide the next instruction to be executed instead of the last retired instruction.

Workaround

There is no workaround available for this erratum.

2287723
The vector register file may not be cleared when tail chaining from a secure to non-secure exception

Status

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

Description

If a late arrival or derived exception causes a tail chain from secure to non-secure mode, the partially restored FP state may not be cleared under some circumstances.

Configurations affected

This erratum affects all configurations with FP or MVE configured which allow the EPU to be placed in a retention power mode.

Conditions

This erratum occurs when all the following conditions are met:

- EPU power domain is configured for retention
- Software writes FPCCR_NS.LSPACT to 1 while executing a secure exception handler. (FPCCR_S.S==0, FPCCR_S.LSPACT==0, FPCCR_NS.LSPACT==1)
- An exception return is taken to secure state with an extended stack frame (EXCRET.FType==0)
- During the exception return, a late arrival or derived exception triggers a tail chain from secure to non-secure state

Implications

If this erratum occurs, the partially restored vector register contents may not be cleared.

Workaround

No workaround is required. Erratum conditions are not expected to be met in correct software.

2312664
Under limited circumstances, LDRD or VLDR instructions that access both device and non-device memory can deadlock the memory system

Status

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

Description

A LDRD or VLDR that crosses into or from device memory with a word aligned non faulting access can block a recently executed device store from progressing and thereby creating a deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- A store to device memory is executed and has not yet completed its write to the memory system.
- The next memory access instruction is a LDRD or VLDR instruction that is word aligned, and accesses device memory with one word and non-device memory with the another.
- Instructions must have this condition in the first DW of their access
- The accessed locations are in the AXI or PAHB sections of the memory map

Implications

The memory system is unable to complete the load or store operation.

Workaround

There are two mutually exclusive workarounds. Do not allow LDRD or VLDR accesses to cross MAU regions such that they can access both device and non device memory. Or, Perform a DMB operation before any LDRD or VLDR access that accesses both device and non device memory.

2297046
RTL incorrectly halts as it incorrectly computes vector catch on lockup entry

Status

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

Description

If a lockup occurs during an interrupt sub routine when vector-catch is enabled on exception currently being processed, a vector-catch might be triggered on lockup entry.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- Executing an error handler for which the corresponding DEMCR.VC_*ERR bit is set
- DHCSR.C_DEBUGEN == 1
- Halting Debug is Allowed for the current security and privilege level
- While executing an instruction, an exception is generated that escalates to lockup

Implications

When vector catch is enabled, an unexpected entry into halt mode may be observed.

Workaround

In many cases a work-around is not required, as a manned debugger is generally chosen to trigger halting debug manually on lockup entry.

For an un-manned debugger, this can be worked around by having the debugger detect this case by comparing the preferred return address and the vector fetch address and then perform the following steps:

1. Save the fault address and fault status register.
2. Save the current value of DEMCR.
3. Disable DEMCR.VC*_ERR.
4. Resume Execution (lockup will be entered).

5. Restore the fault address and fault status registers.
6. Restore the DEMCR.

2335473
A load or store multiple might cause an incorrect ICI value to be stacked in the RETPSR

Status

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

Description

A load or store multiple instruction which is the first instruction following an IT block might cause a zero stacked in the ICI field of the RETPSR when a valid non-zero ICI value would be expected.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- The last instruction in the IT block is a NOP or direct branch instruction
- The next instruction to be executed is a load or store multiple which is interrupted by an asynchronous exception

Implications

When these conditions occur, the stacked RETPSR ICI value might contain zero instead of the valid non-zero value. If this erratum occurs, the only effect would be a restart instead of resume of the affected load or store multiple. No other register or memory corruption will occur.

Workaround

No workaround is required for this erratum.

2374615
FP exception flags in the FPSCR might be set incorrectly after FP context creation following a security state change

Status

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

Description

The exception flags stored in the *Floating-point Status and Control Register* (FPSCR) may be set incorrectly after the FPSCR is cleared as part of an automatic FP context creation. This might occur when FP instructions are used after a security state transition is made without preserving the non-secure FP context after the security state change.

Configurations affected

This erratum affects configurations with FP.

Conditions

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

1. FPCCR_S.ASPEN is set to enable automatic FP context creation in secure state.
2. A FP instruction executes in non-secure state which would set one of the FP exception bits in the FPSCR.
3. A SG instruction executes to transition from non-secure to secure state.
4. A FP instruction executes in secure state which causes an automatic FP context to be created.

Implications

If this erratum occurs, the FP exception flags might be set by the older FP instruction after the new context has been created and cleared the FPSCR. This sequence is not expected to occur in correct software which would preserve the non-secure context prior to executing FP instructions in secure state.

Workaround

This erratum will not affect proper software sequences and no workaround is expected to be necessary for this erratum.

2374006
Undefined 16bit opcodes could raise a NOCP UsageFault instead of an UNDEFINSTR UsageFault

Status

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

Description

UNDEFINSTR UsageFaults are raised by the core for unallocated instruction encodings or instruction encodings which become `UNDEFINED` or `CONSTRAINED UNPREDICTABLE` due to the opcode encoding. Under some circumstances, these undefined instructions could raise a NOCP UsageFault instead of an UNDEFINSTR UsageFault.

Configurations affected

This erratum affects configurations with MVE.

Conditions

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

1. A VCTP is fetched by the core
2. Specific micro-architectural conditions occur
3. An undefined 16bit instruction is executed by the core

Implications

If this erratum occurs, the NOCP flag instead of the UNDEFINSTR flag could be recorded in the UFSR (UsageFault Status Register) when the UsageFault exception is taken. This erratum will only affect sample silicon and is assessed as Category C for that reason.

Workaround

No workaround is required for this erratum.

2374269
Unprivileged debugger access allowed to read or write the DPDLPSTATE register

Status

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

Description

The Implementation control register, DPDLPSTATE, controls the minimum power modes permitted to be requested by the processor. It should not be accessible by an unprivileged debugger. Due to this erratum, the register can be written or read by an unprivileged debugger access.

Configurations affected:

This erratum affects all configurations.

Conditions

This erratum occurs when the following condition are met:

- The PDDEBUG domain is on and the DBGCLK is enabled.
- A write or read access is performed through an unprivileged DAP request to the implementation control register, DPDLPSTATE, at address 0xE001E304.

Implications

An unprivileged debugger can override the software defined value of the DPDLPSTATE register. The unprivileged debugger may prevent the processor from requesting a low power state when conditions would permit it to do so. An unprivileged debugger is also allowed to read the minimum low-power state permitted by the processor.

Workaround

No workaround is available for this erratum.

2371473
The execution priority might be incorrect for a cycle when an IRQ's priority is updated in its handler

Status

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

Description

The execution priority might be incorrect for a cycle when an IRQ's priority is updated by writing to NVIC_IPRn register when in the IRQ handler. This might occur when the priority update is immediately followed by a security state change and another security state change after two cycles.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum might occur when the following sequence of conditions are met:

1. In IRQ handler write to register NVIC_IPRn to update its own priority.
2. Security state change in the following cycle.
3. Another security state change after two cycles.

Implications

If this erratum occurs, NVIC might not use the updated IRQ priority for pre-emption calculation immediately after the NVIC_IPRn register write for a small window.

Workaround

No workaround is available for this erratum.

2397109
An Instruction might be incorrectly sent to Embedded Trace Macrocell as partially completed instruction

Status

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

Description

Multi-cycle instruction might be sent to Embedded Trace Macrocell (ETM) as a partially completed instruction even if it is not architecturally executed, RETPSR.ECI and RETPSR.ICI is zero and the instruction will be restarted if an exception is taken.

Configurations Affected

This erratum affects configurations with ETM=1.

Conditions

This erratum occurs when multi-cycle instruction like LSM encounters an exception and RETPSR.ECI and RETPSR.ICI is zero.

Implications

If this erratum occurs:

- ViewInst start logic might be triggered by an instruction which is not architecturally executed if *Data Watchpoint and Trace* (DWT) CMPMATCH is set for the instruction.
- Trace stream might incorrectly indicate an instruction as partially completed instruction before an exception is taken even if it is not architecturally executed.
- This erratum does not incorrectly trigger single-shot.

This erratum is not believed to be cause of material problem for sample silicon.

Workaround

No workaround is available for sample silicon.

2311453
Flaw in memory system power down may result in data corruption

Status

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

Description

After a reset the caches are automatically invalidated. If there is no instruction cache or the invalidation of the instruction cache completes before the invalidation of the data cache, then the memory system can indicate it is ready to power down while the data cache invalidation is still ongoing.

Configurations Affected

This erratum affects all configurations with a data cache included. Configurations with a data cache but without an instruction cache have an increased chance of the memory system falsely indicating its low-power readiness.

Conditions

This erratum takes effect when the automatic invalidation of the data cache is active and the memory system is otherwise quiescent (i.e., there are no ongoing loads, stores, or other memory operations). While it is possible that a power down request can occur before any loads are performed, this is not expected.

Implications

The memory system can indicate that it is ready to be powered down while it is actively invalidating the data cache tag RAMs. When powered back up, the data cache may be in an unknown state unless a new full invalidation is performed. The processor may resume invalidation of the cache when powered up and report that it is low power ready before the invalidation completes.

This only occurs when no loads are performed before the power down requests occurs which is not expected.

Workaround

Perform a load operation to any address after reset and before entering low power mode.

2282154
Under limited circumstances, a load to a normal non-cacheable location might not be properly ordered by a barrier instruction

Status

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

Description

A DMB or DSB fails to clear the buffered load data and leads to future load of stale data from non-cacheable normal memory region.

Configurations Affected

This erratum affects all configurations include data cache.

Conditions

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

- Specific microarchitectural conditions occur.
- A DSB or DMB is executed after an LDM.
- No load operation to the AXI memory region that is outside of the 32-byte granule accessed by LDM
- A load that falls to the same 32-byte granule as the LDM.

Implications

The load might get stale data if the memory region is used to pass the shared data.

Workaround

No workaround is expected to be required. If a workaround is required, enacting the setting of ACTLR_S[16] will eliminate this condition, but this setting will impact the performance of non-cacheable operations.

2272772
Debug read to FPSCR may return wrong default value

Status

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

Description

When using the debugger to read FPSCR when FPSCR is not present, the value 0x0004000 may be returned instead of 0x00000000.

Configurations affected

- MVE=0 and FPU = 0

Conditions

This erratum occurs when all the following conditions are met:

- Configuration: MVE=0 and FPU = 0
- Halted and DCRSR is used to read data with select 0b0100001

Implications

When reading from this register, a debugger may see LTPSIZE as "4", which is the inactive value, instead of reading "0" (due to the register not being present). This has no impact on core state or execution. This value is static and cannot result in data leakage, other than the presence or absence of this errata.

Workaround

The debugger should not attempt to read this register when it is non-existent, as determined by the feature id registers.

2276413
Debug writes to privleged only bits of current state using DCRSR may not occur if UDE is enabled

Status

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

Description

When the DAP uses the DCRSR to trigger a write to the state registers for the non-secure register bank, the write to privileged only bit might not occur.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs when all the following conditions are met:

- The debugger has access to privileged non-secure state (NSUIDE = 0) and HaltingDebugAllowed()
- The debugger does not have access to privileged secure state due to SUIDE = 1
- 0b0100011 or 0b0010100 is used in the DCRSR to trigger a transfer to the current state while core is Halted

Implication

- Privileged only state bits in the non-secure bank of the CONTROL register may ignore writes from a properly authorized external debugger.
- This categorization factors in that the releases the erratum applies to will only be used for samples.

Workaround

Avoid setting SUIDE = 1 when debugging non-secure privileged state. This means either allowing full access to the debugger, or just non-secure access.

2365142
Direct cache access to Instruction cache data RAM cannot read half of the RAM locations

Status

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

Description

When using DCAICLR/DCAICRR pair to read Instruction cache data RAM, half of the RAM locations cannot be read.

Configurations affected

This erratum affects all configurations.

Conditions

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

1. Program DCAICLR to read from Instruction cache data RAM (bit[0] = 1) and such that bit[3:2] is 0b10 or 0b01
2. Load from DCAICRR register

Implications

If this erratum occurs, half of the Instruction cache data RAM locations cannot be read out using DCA operations.

Workaround

There is no workaround.

2355269
A DWORD-sized store access to ITM_CIDR2 and ITM_CIDR3 can fault incorrectly on the access to the ITM_CIDR3 when the store is not privileged

Status

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

Description

A DWORD-sized store access to the final two IPPB *Instrumentation Macrocell* (ITM) registers, ITM_CIDR2 and ITM_CIDR3, will fault on the access to ITM_CIDR3 if the store is not privileged. Unprivileged accesses to ITM_CIDR3 can only fault if the Main Extension is not implemented. The Main Extension is implemented for the Cortex-M85.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met:

- An aligned DWORD size store to ITM_CIDR2 and ITCM_CIDR3 (Addresses 0xE0000FFF - 0xE0000FF8).
- The store is not privileged.

Implications

If this erratum occurs, the store will fault on its upper lane access to ITCM_CIDR3 when no fault should occur. However, the store does not need to succeed since both ITM_CIDR2 and ITM_CIDR3 are read-only and a successful store would not alter their values.

Workaround

Only perform WORD-sized store operations to ITM_CIDR2 and ITM_CIDR3 instead of writing them together with one instruction.

2374351
Trace fails to update PC on Imprecise bus fault under IESB fault escalation

Status

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

Description

The *Data Watchpoint and Trace* (DWT) comparators can be set up to match on an Instruction address and programmed to generate a Debug event. This erratum causes watchpoint hit or CMPMATCH to be missed on Instruction address programming on the faulted Instruction address.

Configurations Affected

This erratum affects all configurations.

Conditions

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

1. The DWT Comparator n is implemented, where n = 0-7
2. DWT_COMPn supports Instruction Address matching and is programmed to match on Instruction address
3. DWT_COMPn is programmed to generate debug event action on a match
4. AIRCR.IESB==1, enabling *Implicit Error Synchronization Events* (IESBs)
5. Instruction execution begins for a PC that should match for DWT_COMPn where the instruction is load with the PC as a destination register. An imprecise bus fault is synchronized before the Instruction address matching instruction can complete

Implications

If this erratum occurs:

- An Instruction address watchpoint in the presence of an imprecise bus fault synchronized by an IESB can be missed
- A CMPMATCH to *Embedded Trace Macrocell* (ETM), *Cross Trigger Interface* (CTI), and *Performance Monitoring Unit* (PMU) can be missed

Workaround

No workaround is available for sample silicon.

2631240
Unexpected writes to DEBRs can potentially stop the outstanding linefills from completion

Status

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

Description

When all the ways of a data cache set are subscribed by outstanding linefills or blocked by a valid *Data Cache Error Bank Register* (DEBR), software writes to set DEBR valid for the same set, can prevent the linefill from completion.

Configurations affected

This erratum affects all configurations that include data cache.

Conditions

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

1. Specific microarchitectural conditions occur, all outstanding linefills target the same cache set (S)
2. The sum of linefills and number of valid DEBRs that points to the same set as the linefills is equal to 4
3. Software writes to another DEBR, sets bit[0] to 1 and bits[13:5] to match cache set S

Implications

The processor might hang.

Workaround

There is no workaround. The erratum conditions are not expected to be met in typical software.

The expected usage on DEBRs is that when *Error Correcting Code* (ECC) is detected, hardware will update the register. Only Secure privilege software can read from the register and write to lock or clear the register. It should not set DEBR to valid with random set/way.

2601889
Under limited circumstances, an ECC error on data cache RAMs can lead to data corruption

Status

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

Description

When there are data cache *Error Correcting Code* (ECC) errors detected on a LDR or *Preload Data* (PLD) instruction, there is a small time window that an older store and a PLD can fill the same cache line into different ways, leading to data corruption.

Configurations affected

This erratum affects all configurations that include the data cache and ECC protection for the data cache.

Conditions

This erratum occurs when the following sequence of conditions is met:
1. An older cacheable store to address A cache line L is allocated into the store buffer
2. A younger PLD from the same cache line L is issued
3. The PLD misses the data cache and encounters an ECC error or a younger load encounters an ECC error on its cache lookup

Implications

If this erratum occurs, in the presence of data cache RAM ECC errors, two copies of the line can be filled into the cache, causing data corruption.

Workaround

There is no workaround proposed.

2593278
Under limited conditions, an imprecise abort may not be reported when a tag RAM ECC error is detected and data loss has occurred

Status

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

Description

Under certain conditions, an *Error Correcting Code* (ECC) error in the data cache tag RAM can prevent an imprecise abort from being reported by the memory system to indicate data loss. This can occur either when:

- A store is hazarding an ongoing linefill and also encounters an ECC error when it looks up the data cache tag RAM. The linefill that the store is hazarding subsequently encounters a DECERR or SLVERR. (Variant 1)
- The ECC maintenance FSM gets a fatal ECC tag error while handling a different ECC error encountered by a load, PLD, or store. (Variant 2)

Configurations affected

This erratum affects all configurations that include the data cache and ECC protection for the data cache.

Conditions

This erratum occurs when one of the following sequences of conditions is met:

Variant 1 - Tag error on a store hazarding an aborting linefill

1. ECC must be enabled and remain enabled through all following steps
2. A store S must read the data cache tag RAM and/or data RAM and encounter any kind of tag ECC error. Store S is one of at least two stores in the store buffer that are writing the same cacheline but different doublewords in that cacheline
3. Store S hazards an ongoing linefill for its cacheline
4. While store S is waiting for the ECC error from step 2 to be resolved, the linefill it hazards gets a DECERR or SLVERR while retrieving data from main memory
5. Store S or one of the other stores matching its cacheline in the store buffer discards its data. An imprecise abort is not reported to indicate the loss of data.

Variant 2 - The ECC maintenance FSM gets a different ECC error than the original request

1. ECC must be enabled and remain enabled through all following steps
2. A load, PLD, or store must read the data cache tag RAM and/or data RAM and encounter an ECC error.
    - The ECC error is one of the following:
        - A correctable tag error
        - A correctable data error
        - A fatal data error, but only if bus poisoning is enabled (MSCR.EVECCFAULT == 0)
    - The cacheline in which the ECC error is detected is dirty
3. The ECC maintenance FSM is started to handle the ECC error detected in step 2
4. The ECC maintenance FSM encounters a fatal tag error when it reads the data cache tag RAM.
5. The fatal tag error indicates that the state of the cacheline is not known, so the ECC maintenance FSM invalidates the dirty line causing data loss. An imprecise abort is not reported to indicate the loss of data.

Implications

In both variants, all ECC errors are reported correctly on the Error Interface signals and to the *Reliability, Availability, and Serviceability* (RAS) registers, but an imprecise abort is not reported to indicate that data has been lost. The core will continue execution assuming that the cache is intact until an imprecise abort is raised for some other reason.

Workaround

There is no workaround.

2589246
An incorrect fault address is reported for an unprivileged aligned doubleword store to the STIR and RFSR registers when CCR.USERSETMPEND is set

Status

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

Description

A non-vector doubleword store aligned to the address of the STIR or STIR_NS writes two registers: the STIR/STIR_NS and the RFSR/RFSR_NS. If the store does not have privileged access and CCR.USERSETMPEND is set, then the write to the RFSR/RFSR_NS will fault but the write to the STIR/STIR_NS will not fault. When this occurs, the fault address reported for the RFSR/RFSR_NS write that faulted will be one word above the RFSR/RFSR_NS instead of the register address itself.

Configurations affected

This erratum affects all configurations.

Conditions

All of the following conditions must occur in order for this erratum to take effect:

- The core must execute a doubleword store to the STIR and RFSR or STIR_NS and RFSR_NS
- The store must be doubleword aligned
- The store cannot be a M-profile Vector Extension (MVE) instruction
- The store cannot have privileged access
- CCR.USERSETMPEND must be set at the time the store executes

Implications

The BFAR is updated to the address one word above where the actual fault occurred, even though that address was never actually accessed.

Workaround

Normal software is not expected to write the STIR or STIR_NS using doubleword stores. The erratum may be avoided by using only word-sized stores to write the STIR and STIR_NS registers.

2902253
ITM can generate synchronization and timestamp packets when NIDEN=0

Status

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

Description

The *Instrumentation Trace Macrocell* (ITM) generates synchronization and timestamp packets based on *Data Watchpoint Trace* (DWT) signaling if trace is enabled, Non-secure Non-invasive debug is allowed, and synchronization packets transmission or timestamp packet generation is enabled respectively. Due to this erratum, the ITM will incorrectly generate Synchronization and Local and Global Timestamps packets even when Non-secure Non-invasive debug is not allowed, if the remaining conditions are met.

Configurations affected

This erratum affects all configurations of the processor configured with ITM = 1.

Conditions

This erratum occurs when the following conditions are met:

- Global enable for all DWT, PMU, and ITM features is set, that is, DEMCR.TRCENA=1, AND
- Non-Secure Non-Invasive debug is not allowed, AND
  - Synchronization packet transmission for a synchronous TPIU is enabled i.e. ITM_TCR.SYNCENA=1, OR
  - Local timestamp generation is enabled, OR
  - Global timestamp generation is enabled.

Implications

If this erratum occurs, the ITM will generate some trace even when trace generation is not allowed due to Non-invasive Non-secure debug not being enabled. However, as the packets being generated in this case are synchronization and timestamp packets and these packets do not represent security/functional insight into the software running on the PE in restricted modes, there are no security implications of this erratum.

Workaround

There is no workaround for this erratum.

2666660
An unusual combination of events can cause the core to hang instead of lock up on SOE entry

Status

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

Description

When an LSERR is triggered at the same time as a sleep-on-exit, if fault-mask is set, the lockup state entry may conflict with the sleep on exit, which results in neither sleep nor lockup entering successfully.

Configurations affected

This erratum affects all configurations with FP or MVE greater than 0.

Conditions

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

1. Fault mask is set and boost priority such that a lockup on a fault would be required.
2. SLEEPONEXIT is set.
3. An exception return to thread mode which will trigger sleep on exit occurs.
4. The exception return generates the following fault:
    - LSERR due to ~EXCRETURN.ftype & ~EXCRETURN.s & FPCCR_S.lspact

Implications

Lockup is expected to occur, but instead the core will wait for the next interrupt, without sleeping. This may cause additional power draw. Additionally, the fault may not be handled in a timely manner.

Workaround

No workaround is required in properly functioning software, as the circumstances can only be created by secure privileged code erroneously altering either EXCRETURN or FPCCR.

2674410
An AXI load reading some portion of a word that an older store is writing might fail to report a BusFault in certain situations

Status

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

Description

When a load targeting AXI memory is reading at least part of the same word that an ongoing older store is writing, the load might fail to report a BusFault that it encounters on its read. This only occurs in one of two situations:

1. Non-ECC case: The AXI load is reading from Non-cacheable memory. An older store or series of older stores (store A) is writing all the same bytes that the load is reading in a particular word, but then a second store (store B) is writing fewer bytes in the word than the load is reading. Store A is older than store B, but both stores are older than the load.

2. ECC case: An older store or series of older stores is writing all the same bytes that the AXI load is reading in a particular word. The store or series of stores must be draining from the store buffer on the same cycle that the load looks up the data cache. The AXI load must then get a tag or data RAM ECC error on its data cache lookup.

Configurations affected

This erratum affects all configurations. Configurations which include ECC have an increased probability of this erratum taking effect since such configurations allow for the erratum to take effect in a second way (the ECC case described previously).

Conditions

This erratum occurs when one the following sequences of conditions is met:

Sequence A - Non-ECC Case
1. A store or series of stores (Store A) writes some or all the bytes in word W in AXI memory.
2. Another store (Store B) writes some portion of word W also, but fewer bytes than store A is writing.
3. A Non-cacheable load is reading the same number of bytes from word W as store A is writing.
4. Store A completes its write to AXI memory.
5. Store B completes its write to AXI memory.
6. The load requests an AXI memory read.
7. The load's AXI memory read gets a BusFault, either due to a SLVERR or DECERR response from AXI or because the AXI data has been poisoned.

Sequence B - ECC Case

1. A store or series of stores writes some or all the bytes in word W in AXI memory.

2. A load is reading the same number of bytes from word W that the store is writing.

3. The load looks up the data cache on the same cycle that the store is draining from the store buffer.

4. The load must get either a tag or data RAM ECC error on its data cache lookup.

5. The data cache handles the ECC error by invalidating (or cleaning and invalidating in the case of a dirty cacheline) the erroneous cache entry.

6. The load requests an AXI memory read.

7. The load's AXI memory read gets a BusFault, either due to a SLVERR or DECERR response from AXI or because the AXI data has been poisoned.

Implications

The load fails to report a BusFault from its AXI read response. The load might return erroneous data as a result of the fault not being taken.

Workaround

There is no workaround.

2666664
Under limited conditions, an unaligned MVE load reading the TCMs may return corrupted data if it encounters a transient BusFault on its TCM read

Status

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

Description

An unaligned *M-profile Vector Extension* (MVE) load reading a *Tightly Coupled Memory* (TCM) can return corrupted data for the third word it is accessing when the TCM read for that word encounters a transient BusFault. Data corruption only occurs if an older store is writing some (but not all) of the bytes in this word, the bytes in the word that the store is not writing are predicated out by the MVE load, and the TCM accesses for the store and load occur with a particular timing between them.

Configurations affected

This erratum affects all configurations that include the MVE and use TCM wait states on accesses to at least one of the TCMs. Configurations which also include *Error Correcting Code* (ECC) have an increased probability of this erratum taking effect since those configurations add another possible transient BusFault (due to a fatal ECC error on a TCM read).

Conditions

This erratum occurs when all the following conditions are met:

- This erratum only occurs for unaligned MVE loads reading one or more of the TCMs
- All bytes in the third word read by the unaligned MVE load must either be predicated out or be forwarded from an older store. However, this word cannot be fully predicated or fully forwarded. There must be some byte in the word that is only predicated and another byte which is only forwarded.
- The TCM read for the third word accessed by the unaligned MVE load must stay at least one cycle in a TCM wait state
- The TCM read for the third word accessed by the unaligned MVE load must not hit a TEBR register
- The TCM read for the third word accessed by the unaligned MVE load must not complete until the older store forwarding data to the load for the same word has finished its TCM write
- The TCM read for the third word accessed by the unaligned MVE load must get a transient BusFault, either due to a TCMERR or a fatal ECC error. The older store forwarding data to the load for this word must not have seen the BusFault earlier.

Implications

The unaligned MVE load will return data for the third word it accessed from a TCM read that faulted. The data may be corrupted due to the BusFault and the BusFault is not reported. In the case of a fatal ECC error from the TCM, the DME bus will still report the error and the RAS registers will be updated for the ECC error correctly.

Workaround

There is no workaround.

2746551
Store address to DWT incorrect for certain VSTR operations

Status

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

Description

The *Data Watchpoint and Trace* (DWT) comparators can be set up to match on a data address match, and based on the action programming can generate a debug event to the core or trace packets to ITM. This erratum causes watchpoint hits to happen or be missed for certain non-doubleword crossing *Vector Store* (VSTR) operations that are dual issued with another operation, resulting in unreliable debug event generation or trace packet generation on data address programming around the address of the VSTR operation.

Configurations Affected

This erratum affects all configurations of Cortex-M85.

Conditions

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

1. The DWT Comparator n is implemented, for DBGLVL = 1, n =1 and DBGLVL =2, n = 1 or 3
2. DWT_COMPn supports Data address matching and is programmed to match on store address
3. DWT_COMPn is programmed to generate debug event action or trace packet generation on a match with the store address of interest
4. A non-doubleword crossing VSTR operation is dual issued with another operation

Implications

Due to this erratum, the following might occur on store data address matching in the DWT:

- A false watchpoint hit can be reported back to the core
- A watchpoint hit can be missed
- A false CMPMATCH can be triggered/missed to ETM
- Unreliable MATCH or PC VALUE MATCH packet generation

Workaround

To work around this erratum, set up matching at 8byte boundaries within the VSTR of interest, which might generate false hits to occur on the specific addresses.

2757159
Under limited circumstances, store operations do not update the data cache correctly

Status

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

Description

A store operation to an address that is resident in the cache writes to external memory without updating the data cache.

Configurations affected

This erratum affects all configurations that include data cache.

Conditions

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

1. Cacheable memory is attributed as No Write-Allocate or write streaming is enabled (Secure version of ACTLR.DISNWAMODE is set to 0)
2. An older store followed closely by a younger MVE load, the load is crossing between normal memory and device region, and accessing different double word in the same cache line as the store.
3. Under specific microarchitectural conditions, store writes to external memory without updating the data cache

Implications

It is not expected that software will have incremental MVE load cross between normal and device regions. If this erratum occurs, the cache line may be corrupted and subsequent read of an address within this cache line may return stale data.

Workaround

No workaround is required.

2682779
After deactivating the instruction cache, self-modified code might not be executed correctly

Status

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

Description

Cortex-M85 supports the use of self-modifying code but requires the following before the modified code is executed:
1) Cache maintenance operations are performed to flush the old code from the instruction cache.
2) An ISB instruction is executed to flush old code from buffers internal to the processor.

Cortex-M85 also allows the instruction cache to be deactivated using the IACTIVE bit of *Memory System Control Register* (MSCR). The instruction cache is usually deactivated to allow power down of the cache SRAM.

If the instruction cache is deactivated shortly before code is modified the old code might be executed despite the use of cache maintenance operations and the execution of an ISB instruction.

Due to this erratum, an internal instruction buffer is not flushed when the instruction cache is inactive. This means old code can be erroneously executed after the associated address has been modified by software or from a debugger using the appropriate cache maintenance and synchronization operations.

Configurations affected

This erratum affects configurations of the Cortex-M85 processor that includes the instruction cache. Configuration parameter ICACHESZ > 0.

Conditions

1) There is a cache line containing executable code and which has memory attributes indicating that it is cacheable.
2) The cache line is in normal memory and not in *Instruction Tightly Coupled Memory* (ITCM) and *Data Tightly Coupled Memory* (DTCM).
3) The cache line is read from external memory due to instruction fetch.
4) A store is executed that writes to the memory location corresponding to the address which has already been fetched into the instruction cache
5) The write modifies instruction data and the data is cleaned out to L2 memory using the code sequence in the Arm v8-M Architecture Reference Manual (see identifier IMLLC).
6) The instruction cache is active when the cache line is read from external memory.
7) The instruction cache is inactive when the required cache maintenance operations occur.
8) The instruction cache is inactive when the required ISB instruction is executed.
9) Between the write and the execution of the modified instruction no other cache line is read from external memory due to instruction fetch.

Note: A cache line is an area of memory 32 bytes in size and 32 byte-aligned.

Implications

If the conditions listed in this erratum occur the previous code at the instruction location will be executed from the internal buffer until it is replaced with new instructions.
The erratum is unlikely to result in any real issues in a realistic software or debug scenario as executing code from an inactive instruction cache will result in very low performance. When the cache is inactive to minimize power Arm recommends running code from the *Tightly Coupled Memory* (TCM).

Workaround

No workaround necessary as this issue is unlikely to be encountered in a realistic software or debug scenario.

2677972
CTITRIGIN not connected to DWT_CMPMATCH for ETM = 0 configuration

Status

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

Description

The *Cross Trigger Interface* (CTI) component enables the processor debug logic and *Embedded Trace Macrocell* (ETM) to interact with each other and with additional CoreSight debug and trace components in the system using trigger events across a standard interface and protocol. This allows software running on Cortex-M85 to be debugged efficiently in a larger system containing multiple processors using DWT CMPMATCH as one of the trigger sources. This erratum causes loss of DWT triggering functionality for configurations with ETM = 0.

Configurations Affected

This erratum affects Cortex-M85 configurations with ETM = 0.

Conditions

This erratum occurs when the following conditions are met:

- Cortex M-85 is configured with parameter DBGLVL = 1 or 2 and ETM = 0.

- DWT Comparator n where n < 4 is programmed for a match and CMPMATCH generation is possible.

Implications

DWT_CMPMATCH is not connected to CTI input triggers, causing loss of cross triggering functionality via DWT_CMPMATCH for this configurations.

Workaround

No workaround is available.

2705117
L1D_CACHE_REFILL and L1D_CACHE_MISS_RD PMU events might be inaccurate

Status

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

Description

The following *Performance Monitoring Unit* (PMU) events supported by the Cortex-M85 processor might trigger too frequently:

- L1D_CACHE_REFILL
- L1D_CACHE_MISS_RD

Configurations affected

This erratum affects configurations of the Cortex-M85 processor with the Verilog parameters DBGLVL != 0 and DCACHESZ != 0.

Conditions

There are no special conditions associated with this erratum.

Implications

L1D_CACHE_REFILL

L1D_CACHE_REFILL is incorrectly triggered for data cache refills caused by PLD instructions and the data prefetcher. Therefore, this can cause the frequency for this event to be higher than the frequency of the L1D_CACHE event. This can result in an inaccurate value for data cache refill rate ratio calculations.

L1D_CACHE_MISS_RD

L1D_CACHE_MISS_RD is incorrectly triggered for data cache misses due to PLD instructions, store instructions, and linefills made by the data prefetcher. Therefore, this can cause the frequency for this event to be higher than the frequency of the L1D_CACHE_RD event. This can result in an inaccurate value for data cache read operation miss rate ratio calculations.

Workaround

There is no workaround for this erratum.

2614056
Under limited conditions, a doubleword-aligned doubleword ITCM load can read data corrupted by an ECC error

Status

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

Description

A doubleword-aligned doubleword *Instruction Tightly Coupled Memory* (ITCM) load makes separate ITCM requests for each word of data it reads. If the load's upper word ITCM request enters the response phase while the lower word's ITCM request is stalled in the address phase waiting on an older *Data Tightly Coupled Memory* (DTCM) load to complete, then an *Error Correcting Code* (ECC) error in the ITCM data for the upper word of the load may not be detected.

Configurations affected

This erratum affects all configurations that include ECC protection for the ITCM.

Conditions

This erratum occurs when all the following conditions are met:

- During the address phase of the ITCM load there must be an older DTCM load in its response phase with **TCMWAIT** HIGH for the DTCM it is reading
- The load must be reading a doubleword from the ITCM
- The load must be doubleword-aligned
- The data return for the upper word of the ITCM load must get an ECC error

Implications

The load returns incorrect data. Program flow might be altered depending on how software uses the value read. 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.

2640876
Load data value for certain unaligned vector loads incorrect to DWT for data value matching

Status

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

Description

The *Data Watchpoint and Trace* (DWT) comparators can be set up to match on a Data Value match, and based on the action programming can generate a debug event to the core. This erratum causes watchpoint hits to happen or be missed for unaligned vector load operation, resulting in unreliable debug event generation on Data Value match programming in the DWT

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met:

- The DWT Comparator n is implemented, for DBGLVL = 1, n =1 and DBGLVL =2, n = 1 or 3.
- DWT_COMPn supports Data Value matching and is programmed to match on Load Data Value.
- DWT_COMPn is programmed to generate debug event action on a match.
- A data value comparator match is expected on an unaligned vector load operation.

Implications

- A false watchpoint hit can be reported back to the core.
- A watchpoint hit can be missed.
- A false CMPMATCH can be triggered/missed to ETM.

Workaround

- No workaround is available.

2651727
Unprivileged debug transactions to the STIR and EVENTSPR registers can pend interrupts

Status

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

Description

Due to this erratum, an unprivileged debugger access to the architectural STIR register can pend an interrupt (IRQ) and an unprivileged debugger access to the **IMPLEMENTATION DEFINED** EVENTSPR can pend a *non-maskable interrupt* (NMI). Writes to these two registers from an unprivileged debugger should be ignored and a fault should be returned to the debugger.

Configurations affected

This erratum affects all configurations of Cortex-M85 with Halting debug included, DBGLVL > 0.

Conditions

The erratum part concerning STIR register occurs when all the following conditions are met:

- DAUTHCTRL.UIDAPEN is set to 0b1
- The CCR.USERSETMPEND bit is 0b0
- A debugger performs an unprivileged byte or halfword access to the second byte of the STIR register.

The erratum part concerning EVENTSPR register occurs when all the following conditions are met:

- DAUTHCTRL.UIDAPEN is set to 0b1
- A debugger performs an unprivileged access to EVENTSPR

Implications

The implications of unprivileged debugger accesses to the STIR register under the conditions listed above are as follows:

- A byte or halfword read/write that is not word-aligned will not fault.
- A byte write to address 0xE000EF01 can pend interrupt 0 or interrupt 256 depending on bit[0] of the written byte.

The implications of unprivileged debugger accesses to the EVENTSPR register under the conditions listed above are as follows:

- A write to the address 0xE001E400 with bit[1] set in the write-data will pend a non-maskable interrupt.

Note:

- There is no possibility of leaking Secure or privileged information to an unprivileged debugger.
- This erratum does not apply to software running on the processor and so the behavior is as specified in the Armv8-M Architecture Reference Manual. Therefore, unprivileged software cannot access the STIR register if it is not permitted to do so by the CCR.USERSETMPEND bit or the EVENTSPR register.
- The STIR and EVENTSPR register are both write-only registers, so this erratum has no implications for unprivileged debugger reads.
- The implications of this erratum can be mitigated by including interrupt service routines for NMI and interrupt 0 and interrupt 256 (if these maskable interrupts are enabled).

Workaround

There is no workaround for this erratum.

2446484
Simultaneous ECC errors on a store tag lookup and a load data read in the data cache can cause the ECC error for the load to not get corrected

Status

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

Description

When a store tag lookup and load data read occur simultaneously in the data cache and both get Error Correcting Code (ECC) errors, the ECC error for the load might not be corrected. The load ECC error is not corrected only if the store and load are accessing the same way in different cache sets.

Configurations affected

This erratum affects all configurations which include ECC protection for the data cache.

Conditions

This erratum occurs when all the following conditions are met:

- ECC must be enabled
- A store is looking up the data cache tag RAM on the same cycle that a load is looking up the data cache data RAM
- The store and load must be accessing different cache sets
- The store must get a tag ECC error in the same way that the load gets a data ECC error

Implications

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. The data ECC error encountered by the load is reported properly but is not corrected. The load will proceed to request a linefill for the errored cacheline, resulting in double-allocation of the line into the data cache.

Workaround

No workaround is required.

2439477
A Secure read of ICSR_NS.VECTPENDING might return the Secure value

Status

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

Description

ICSR.VECTPENDING field might not be masked and return a Secure value when accessed as Non-secure from Secure state.

Configurations affected

This erratum affects all configurations.

Conditions

A Secure exception is the highest priority pending exception and one of the following occurs:

- The Secure PE attempts to read the Non-secure version of Interrupt Control and State Register (ICSR_NS) located at address (0xE002ED04)
- A Secure debugger (DHCSR.S_SDE==1) attempts to read the ICSR_NS (that is DSCSR.SBRSELEN==0 and PE is Non-secure or DSCSR.SBRSELEN==1 and DSCSR.SBRSEL==0)

Implications

The ICSR_NS read will return the unmasked value of the VECTPENDING field. This is only possible from Secure state. No information is leaked to Non-secure.

Workaround

There is no workaround for this erratum.

2445488
ETM incorrectly reports there are also additional serious exceptions when reporting entry into a serious exception

Status

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

Description

The sampling of pending serious exceptions incorrectly includes the current exception transitioning from pending to active.

Configurations affected

This erratum affects configurations with *Embedded Trace Macrocell* (ETM).

Conditions

This erratum occurs when all the following conditions are met:

- An entry to an exception that is deemed to be related to a serious fault occurs.
- There are no other pending exceptions that are related to a serious fault.

Implication

If this erratum occurs, the P-bit may be set in the ETM packet that reports the exception entry to a serious exception. This indicates that there is also a serious exception pending, when in fact the only serious pending exception just transitioned to active.

Workaround

There is no workaround.

2444277
Store data value for certain MVE stores is incorrect to DWT for data value matching

Status

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

Description

The *Data Watchpoint and Trace* (DWT) comparators can be set up to match on a Data value match, and based on the action programming can generate a debug event to the core. This erratum causes watchpoint hits to happen or be missed for *M-profile Vector Extension* (MVE) store operations that follow a fully predicated (fully inactive) MVE Store operation resulting in unreliable debug event generation on Data value match programming in the DWT.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met:

- The DWT Comparator n is implemented, for DBGLVL = 1, n =1 and DBGLVL =2, n = 1 or 3.
- DWT_COMPn supports Data value matching and is programmed to match on store Data value.
- DWT_COMPn is programmed to generate debug event action on a match.
- A Data value comparator match is expected on an MVE store operation following a MVE store that is fully predicated(fully inactive)

Implications

- A false watchpoint hit can be reported back to the core.
- A watchpoint hit can be missed.
- A false CMPMATCH can be triggered/missed to *Embedded Trace Macrocell* (ETM)

Workaround

We recommend using data address matching on the same MVE operation to address some of the implications. No workaround available for the Data value matching issue reported here.

2490920
A malformed tail-predicated loop at the start of a Secure function may use the value of LTPSIZE without masking

Status

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

Description

If there is a *Secure Gateway* (SG) instruction followed by a LETP instruction, without an intervening WLSTP, DLSTP, or CP10/11 operation, and a floating point context is initialized by the LETP instruction due to the SG instruction clearing CONTROL_S.SFPA, then the LETP may use the value of LTPSIZE before the context initialization.

Configurations affected

This erratum affects all configurations that include the *M-profile Vector Extension* (MVE >0).

Conditions

This erratum can occur when:

- FPCCR_S.ASPEN = 1
- FPCCR_S.LSPACT = 0
- LR > 1;
- CPACR_S and CPPWR_S are configured to allow access to CP10
- CONTROL_S.SFPA = 1
- The core is in Non-secure state

And the following sequence of instructions occurs:

1. An SG instruction in a Secure attributed memory location.
2. Fewer than 8 instructions, none of which are subject to automatic floating point context maintenance.
3. An LETP instruction

Implications

If this erratum occurs, the LR value may decrement an incorrect amount or the LETP may branch to its target address incorrectly.

Workaround

No workaround is required. The sequence of instructions in the erratum conditions is not expected to occur in typical software.

2452728

Before RAM power is up and completes cache auto invalidation sequence, i.e. D$ is accessible, a cacheable store data might not be observed by a load to the same address after D$ is accessible

Status

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

Description

When core power is up and starts executing code before the data cache RAM power is up and completes cache auto invalidation sequence, data stored to a cacheable address might not be observed by a younger load to the same address.

Configurations affected

This erratum affects all configurations with data cache.

Conditions

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

1. Core power is up running but RAM power is not up or the cache auto invalidation sequence has not completed
2. A store to cacheable address A belongs to cache line L
3. RAM power is up and cache auto invalidation completes
4. A load from different address in the same cache line L as the store
5. A load from address A

Implications

The load will get stale data if this erratum occurs.

Workaround

There are two potential workarounds for this erratum:
1. First, disable the auto invalidate sequence. This can be done by tying off the input pin INITL1RSTDIS to 1. Then, execute DCISW ops to loop through all sets and ways.
2. SW Poll bit[17] of MSCR register (address is 0E00_1E00), only after it is low, can start normal loads and stores to cacheable memory locations.

2451384
CHAIN PMU event unconnected to PMU EVENTBUS

Status

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

Description

The *Performance Monitoring Unit* (PMU) event CHAIN(0x001E) is defined in the Armv8-M architecture and counts the occurrences for an odd numbered PMU counter, when an overflow occurs on the preceding even-numbered counter on the same PE. This erratum causes the count to remain at 0 even when enabled and never increment.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- The event CHAIN(0x001E) is selected in the PMU Event Type and Filter Register, PMU_EVTYPERn, configuring event counter n
- Event counter n is enabled in PMU_CNTENSET.P[n]

Implications

CHAIN PMU event, when enabled, will not count and will remain at 0.

Workaround

No workaround is available for sample silicon.

3030982
Store data value for certain aligned vector stores incorrect to DWT for data value matching

Status

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

Description

The *Data Watchpoint and Trace* (DWT) comparators can be set up to match on a Data Value store match, and based on the action programming can generate a debug event to the core. This erratum causes watchpoint hits to happen or be missed for aligned vector store operation, resulting in unreliable debug event generation on Data Value match programming in the DWT.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are met:

- The DWT Comparator n is implemented, for DBGLVL = 1, n =1 and DBGLVL =2, n = 1 or 3.
- DWT_COMPn supports Data Value matching and is programmed to match on Store Data Value.
- DWT_COMPn is programmed to generate debug event action on a match.
- A Data Value comparator match is expected on an aligned vector store operation.
- The access is generated by a vector store with a memory size smaller than the element size (for example: vstrb.16 Qd, [Rn], vstrb.32 Qd, [Rn], or vstrh.32 Qd, [Rn]).

Implications

- A false watchpoint hit can be reported back to the core.
- A watchpoint hit can be missed.
- A false CMPMATCH can be triggered/missed to ETM.

Workaround

- No workaround is expected to be required.

2988396
DWT data value packets can be generated while the match packet from same access overflows

Status

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

Description

The *Data Watchpoint and Trace unit* (DWT) can generate match packet, data address, and data value packet for a single instruction if the trace is enabled and programmed accordingly. If an overflow occurs while reporting the match packet, then the other related packets should also not be emitted. Due to this erratum, the DWT can incorrectly report the related data value packets of an overflowed match packet.

Configurations affected

This erratum affects all configurations of the processor configured with ITM = 1.

Conditions

This erratum occurs when the following conditions are met:

- DWT and ITM are enabled for packet generation.
- Two linked DWT FUNCTION registers are programmed with a data address with value and data address limit range check set to generate PC value, data address, and data value trace packets (for example: FUNCTION0.MATCH=0b11xx, FUNCTION0.ACTION=0b11, FUNCTION1.MATCH=0b0111, FUNCTION1.ACTION=0b11).
- The traced access was a double word or unaligned word access.

Implications

In the case where this erratum occurs, the DWT will generate an overflow followed by data value packets without a corresponding match packet. There are no security implications due to this erratum.

Workaround

There is no workaround for this erratum.

3319763
ACTLR_S bit[11] can not disable No Write-Allocate mode

Status

Fault Type: Programmer Category C
Fault Status: Present in r1p0. Fixed in r1p1

Description

When a memory region is marked as Cacheable Write-Allocate, it normally allocates a cache line on a write miss. However, there are some situations where allocating on writes is undesirable. The processor has logic to detect the write streaming mode and disable allocating on writes in write streaming mode. There is an ACTLR bit added to disable the No Write-Allocate behavior and the bit is not working properly.

Configurations affected

This erratum affects all configurations with data cache.

Conditions

This erratum occurs when the following conditions are met:

- Bit [11] of secure version of ACTLR is set before or after the following sequence
- Stores to memory regions attributed as cacheable write-allocate
- Stores are updating the entire cacheline before targeting different line
- Write streaming is detected

Implications

Once the core is in No Write-Allocate mode, we cannot disable the behavior via ACTLR.

Workaround

No workaround is required.

3078268
Incorrect multi-copy atomicity ordering between Non-cacheable loads with forwarded store data

Status

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

Description

Under limited microarchitectural conditions, of the two dual-issued loads, the younger load is observing a store whose coherence order is before the store observed by the older load.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs when all the following conditions are met:

- Two loads are dual-issued
- Memory attributes of the accessed location is Non-cacheable and Shareable
- Older load access size is bigger than the younger one and the loads have overlapping bytes
- Previous store to the same location has not written to the external memory
- The store can forward all data to the younger load, but only partial data to the older load
- The store is ordered before a store from another PE
- The older load observes the store from the other PE

Implications

The younger load observes the older store in the coherence order, the older load observes the younger store.

Workaround

No workaround is required.