arm CHINA

# Cortex®-M52 AT650, Cortex®-M52 with TrustZone AT651 and Cortex®-M52 MVE Extension AT652

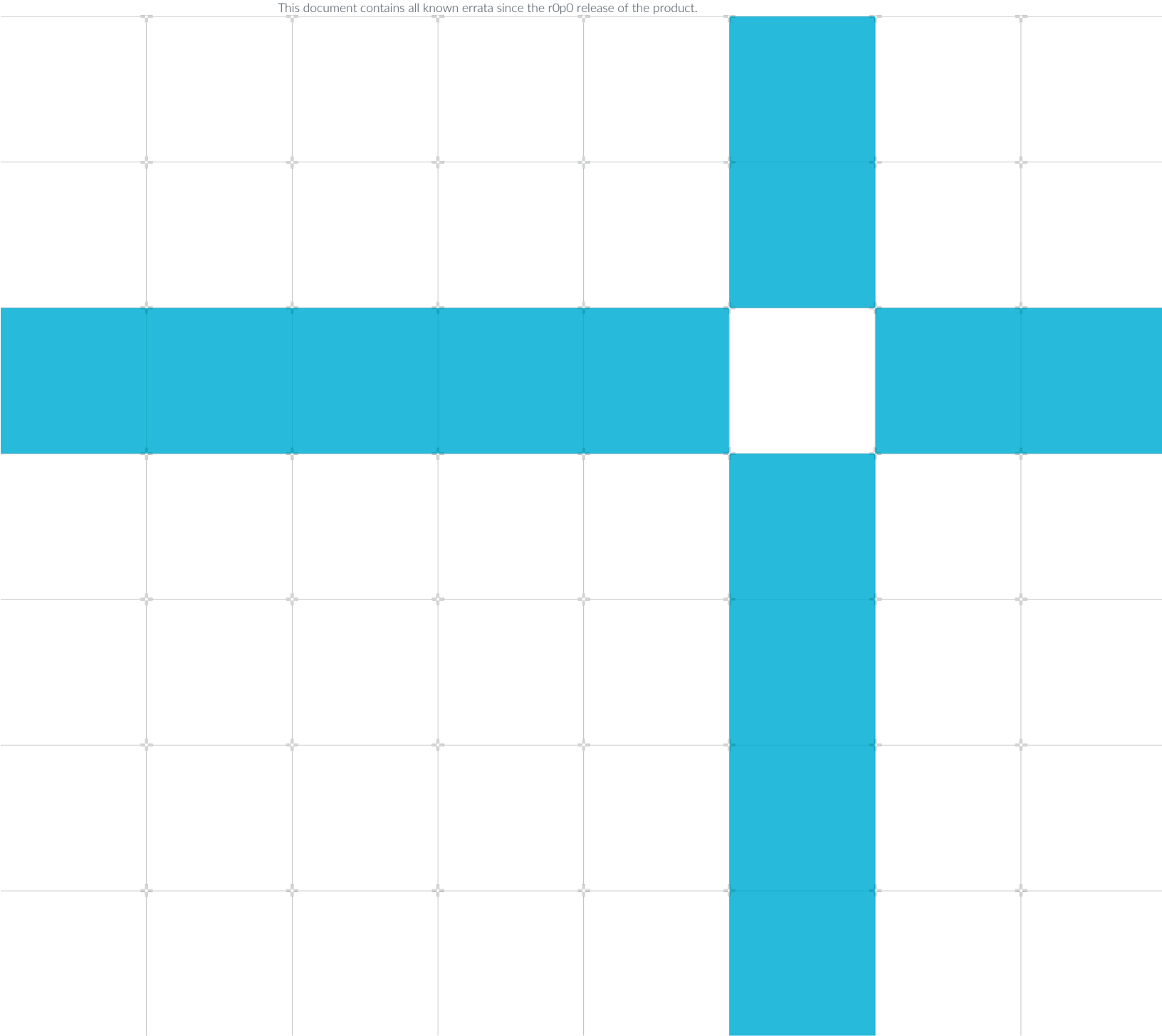
## Software Developer Errata Notice

**Date of issue:** 30-Sep-23



**Document version:** 4.0
**Document ID:** SPD-CPU-AT650-000009

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



## Non-Confidential proprietary notice

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.

Arm Technology (China) Co., Ltd. registered in China.
Room 201, Building A, No. 1 First Qianwan Road, Qianhai Shengang Cooperation Zone, Shenzhen, the People's Republic of China.

(LES-PRE-20349 - Arm China)

## 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 China and the party that Arm China delivered this document to.

Unrestricted Access is an Arm China internal classification.

## Product status

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

## Web address

http://www.armchina.com

## Feedback on this product

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

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

## Feedback on this document

If you have comments on content, send an e-mail to support-team@armchina.com giving:

- The document revision or version.
- The document number: SPD-CPU-AT650-000009.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm China also welcomes general suggestions for additions and improvements.

# Contents

# Introduction

## Scope

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

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

## Categorization of errata

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

| | |
|---|---|
| Category A | A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications. |
| Category A (Rare) | A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification, and usage. |
| Category B | A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications. |
| Category B (Rare) | A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification, and usage. |
| Category C | A minor error. |

# Change control

Errata are listed in this section if they are new to the document, or marked as “updated” if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The errata summary table on page 8 identifies errata that have been fixed in each product revision.

**30-Sep-2023: Changes in document version 4.0**

| ID | Status | Area | Cat | Summary |
|---|---|---|---|---|
| No errata in this document version. | | | | |

**9-Dec-2022: Changes in document version 3.0**

| ID | Status | Area | Cat | Summary |
|---|---|---|---|---|
| C00216080 | New | Programmer | C | PMU event L1D_CACHE is incorrectly implemented in the unified cache |
| C00216383 | New | Programmer | C | VLDR+VDDUP (with implicit LE) overlap and uncorrectable ECC error occurs, core-ETM gives wrong lockup preferred exception return address |

**27-July-2022: Changes in document version 2.0**

| ID | Status | Area | Cat | Summary |
|---|---|---|---|---|
| C00214116 | New | Programmer | C | PMU events, L1D_CACHE, L1D_CACHE_REFILL, L1D_CACHE_RD, L1D_CACHE_MISS_RD, LL_CAHCE_RD and LL_CACHE_MISS_RD, are incorrectly implemented in the data cache or unified cache |
| C00214189 | New | Programmer | C | Unprivileged debug transactions to the EVENTSPR.NMI registers can pend interrupt |
| C00214321 | New | Programmer | C | EPSR.B bit might not set correctly when LDMIA has exact two destination registers and one of them is PC |
| C00214322 | New | Programmer | C | The DWT might generate a Data trace PC value packet with incorrect PC when an SVC instruction is delayed by PPB write and at the same time a new interrupt is invoked |
| C00214380 | New | Programmer | C | I-Cache/U-Cache LFB can't be cleared correctly by CMO or ISB |
| C00214484 | New | Programmer | C | Unprivileged debugger access to DPDLPSTATE register incorrectly modifies register state |
| C00214486 | New | Programmer | C | Floating-point multiple load store instructions which specify unimplemented registers may write to unexpected memory locations and update register incorrectly |
| C00215210 | New | Programmer | C | ST_RETIRED PMU event is inaccurate when a CC failed VMRS FPCXT_NS or FPCXT_S instruction executed |

| ID | Status | Area | Cat | Summary |
|---|---|---|---|---|
| C00215763 | New | Programmer | C | Stepping a vector load/store instruction which cause lockup might not enter debug state or DebugMonitor exception when expected |

**30-April-2022: Changes in document version 1.0**

| ID | Status | Area | Cat | Summary |
|---|---|---|---|---|
| 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 | Cat | Summary | Found in version | Fixed in vision |
|---|---|---|---|---|---|
| C00214116 | Programmer | C | PMU events, L1D_CACHE, L1D_CACHE_REFILL, L1D_CACHE_RD, L1D_CACHE_MISS_RD, LL_CAHCE_RD and LL_CACHE_MISS_RD, are incorrectly implemented in the data cache or unified cache | r0p0 | r0p1 |
| C00214189 | Programmer | C | Unprivileged debug transactions to the EVENTSPR.NMI registers can pend interrupt | r0p0 | r0p1 |
| C00214321 | Programmer | C | EPSR.B bit might not set correctly when LDMIA has exact two destination registers and one of them is PC | r0p0 | r0p1 |
| C00214322 | Programmer | C | The DWT might generate a Data trace PC value packet with incorrect PC when an SVC instruction is delayed by PPB write and at the same time a new interrupt is invoked | r0p0 | r0p1 |
| C00214380 | Programmer | C | I-Cache/U-Cache LFB can't be cleared correctly by CMO or ISB | r0p0 | r0p1 |
| C00214484 | Programmer | C | Unprivileged debugger access to DPDLPSTATE register incorrectly modifies register state | r0p0 | r0p1 |
| C00214486 | Programmer | C | Floating-point multiple load store instructions which specify unimplemented registers may write to unexpected memory locations and update register incorrectly | r0p0 | r0p1 |
| C00215210 | Programmer | C | ST_RETIRED PMU event is inaccurate when a CC failed VMRS FPCXT_NS or FPCXT_S instruction executed | r0p0 | r0p1 |
| C00215763 | Programmer | C | Stepping a vector load/store instruction which cause lockup might not enter debug state or DebugMonitor exception when expected | r0p0 | r0p1 |
| C00216080 | Programmer | C | PMU event L1D_CACHE is incorrectly implemented in the unified cache | r0p0, r0p1 | r0p2 |
| C00216383 | Programmer | C | VLDR+VDDUP (with implicit LE) overlap and uncorrectable ECC error occurs, core-ETM gives wrong lockup preferred exception return address | r0p0, r0p1 | r0p2 |

# Errata descriptions

## Category A

There are no errata in this category.

## Category A (rare)

There are no errata in this category.

## Category B

There are no errata in this category.

## Category B (rare)

There are no errata in this category.

# Category C

## C00214116

## PMU events L1D_CACHE, L1D_CACHE_REFILL, L1D_CACHE_RD, L1D_CACHE_MISS_RD, LL_CAHCE_RD and LL_CACHE_MISS_RD are incorrectly implemented in the data cache or unified cache

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

The following PMU events supported by the Cortex®-M52 processor might be counted incorrectly:

- L1D_CACHE
- L1D_CACHE_RD/LL_CACHE_RD
- L1D_CACHE_MISS_RD/LL_CACHE_MISS_RD
- L1D_CACHE_REFILL

### Configurations affected

This erratum affects the configurations of Cortex®-M52 with data cache or unified cache.

### Conditions

This erratum can occur if data cache or unified cache is included:

- DCACTIVE is 1'b0 and there is cacheable read access.
- There is preload instruction access.
- Cache is inaccessible.

This erratum can occur if data cache is included:

- There is cache miss store access.

This erratum can occur if unified cache is included:

- There is non-code region cacheable read access.

### Implications

L1D_CACHE

- If it is used to gather statistics on the unified cache the number of recorded cache accesses will be higher than expected if there is preload instruction access.

L1D_CACHE_RD/LL_CACHE_RD

- If it is used to gather statistics on the unified cache, the number of recorded cache read accesses will be higher than expected if there is preload instruction access.

L1D_CACHE_MISS_RD/LL_CACHE_MISS_RD

- If it is used to gather statistics on the data cache or unified cache, the number of recorded cache miss read accesses will be higher than expected if MSCR.DCACTIVE is set to 1'b0 during running, or if there is preload instruction access, or when cache is inaccessible.
- If it is used to gather statistics on the data cache, the number of recorded cache miss read accesses will be higher than expected if there is cache miss store access.
- If it is used to gather statistics on the unified cache, the number of recorded cache miss read accesses will be higher than expected if there is non-code region cacheable instruction access.

L1D_CACHE_REFILL

- If it is used to gather statistics on the data cache or unified cache, the number of recorded cache refill accesses will be higher than expected if there is preload instruction access.

## Workaround

There is no workaround provided.

## C00214189

## Unprivileged debug transactions to the EVENTSPR.NMI registers can pend interrupt.

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1

### Description

Due to this erratum, an unprivileged debugger access to the IMPLEMENTATION DEFINED EVENTSPR can pend a Non-Maskable Interrupt (NMI). Writes to this register 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®-M52 with Halting debug, DBGLVL >0.

### Conditions

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 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 unprivileged software can't access the EVENTSPR register.
- The EVENTSPR register is write-only register. Therefore, this erratum has no implications for unprivileged debugger reads.
- The implications of this erratum can be mitigated by including interrupt service routines for NMI.

### Workaround

There is no workaround provided.

## C00214321

## EPSR.B bit might not set correctly when LDMIA has exact two destination registers and one of them is PC

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

If PACBTI is configured and BTI enabled, some branch instructions or load to pc instructions should update EPSR.B bit to check whether the target is a landing pad instruction or not to check its target. LDMIA is also this kind of instruction if r15 is in destination list.

Due to this erratum, when lsu indicate the last uop of the LDMIA cannot be killed and at this time, interrupt comes, then EPSR.B bit cannot be set correctly, and cannot check its target.

### Configurations affected

This erratum affects the configurations with PACBTI where BTI is enabled.

### Conditions

This erratum can occur if:

- PACBTI feature is included and control.bti_en is enabled.
- A load multiple instruction with exact two destinations and one is r15, base Rn should be: "!(Rn = R13 && wback)".
- The load access is word aligned.
- The addr_type is Normal (not function return/exception return).
- While executing the second uop, at the window during which lsu indicate it is non-killable and interrupt comes.

### Implications

EPSR.B bit set and clear happened between a branch instruction and target landing pad instruction. This pair can check whether the branch jumped to the expected target.

For the bug scenario, the EPSR.B bit will not set correctly. If the target address is the entry of malicious software, then no fault can be reported and will execute the evil code.

### Workaround

There is no workaround provided.

## C00214322

## The DWT might generate a Data trace PC value packet with incorrect PC when an SVC instruction is delayed by PPB write and at the same time a new interrupt is invoked

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

The Armv8-M architecture (in rule RSFSC) Specifies the PC value in the Data trace PC value packet on exception entry should be the return address for the exception. This erratum can cause the Data Watchpoint and Trace (DWT) to generate a Data trace PC value packet on exception entry stacking operations with the incorrect PC value when an SVC instruction delayed by PPB write and at the same time a new interrupt is invoked.

### Configurations affected

This erratum affects the configurations of the Cortex®-M52 processor configured with a DWT-DBGLVL>0 and ITM.

### Conditions

This erratum occurs if DWT is configured to generate Data Trace PC value packet and all the following conditions occur:

- A PPB write store instruction followed by an SVC instruction.
- A new interrupt arrives when SVC on the first cycle of retire stage.

### Implications

When the above scenario happens, the PC value of the Data Trace PC value packet is incorrectly point to the SVC's instruction address.

### Workaround

There is no workaround provided.

## C00214380

## I-Cache/U-Cache LFB can't be cleared correctly by CMO or ISB

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1

### Description

Signature 1: Self-modifying code failure if code is in LFB
When MSCR.ICACTIVE is toggled from 1 to 0, the cacheline data in LFB will remain for a while. At this moment if self-modifying code such as software breakpoint insertion is executed on this cacheline, the operation might be failed due to inconsistent data between memory and LFB. Because PE cannot have latest data in memory but only obsolete data in LFB. The inconsistency cannot be resolved by ISB or CMO. But the situation would be gone automatically after another non-cacheable data filled into LFB to replace the original cacheable data.

Signature 2: DCISW and DCCISW cannot invalidate data in LFB
DCISW and DCCISW in the Armv8-M architecture are cache maintenance operations to clear specific cacheline by set/way. However, if the cacheline is in the LFB before allocation to cache memory, it can't be invalidated by DCISW and DCCISW.

### Configurations Affected

Signature 1: Affects the configurations of Cortex®-M52 with instruction cache or unified cache.
Signature 2: Affects the configurations of Cortex®-M52 with unified cache.

### Conditions

Signature 1 can occur if:

- Self-modifying code is performed to address hit in LFB in cacheable, non-TCM region right away after cache is deactivated and before first LFB refill.

Signature 2 can occur if:

- Execute DCISW or DCCISW and the specific set/way of cacheline is still in U-Cache LFB.

### Implications

Signature 1: If this erratum occurs, self-modifying code may fail since modified code is invisible from PE.

Signature 2: If this erratum occurs, DCISW and DCCISW cannot invalidate LFB in U-cache even if set/way matches.

## Workaround

There is no workaround provided.

## C00214484

## Unprivileged debugger access to DPDLPSTATE register incorrectly modifies register state

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

The processor contains an IMPLEMENTATION DEFINED register DPDLPSTATE in the Private Peripheral Bus (PPB) region that specifies the wanted low-power state for the Debug power domain in the processor. This register is only accessible by privileged code or a privileged debugger. Due to this erratum, a write to the DPDLPSTATE register from an unprivileged debugger will return a fault, as expected, but the register state will be incorrectly updated.

### Configurations affected

This erratum affects all configurations of the processor configured with Halting debug, DBGLVL > 0.

### Conditions

This erratum occurs when Unprivileged Invasive DAP Access Enable, DAUTHCTRL.UIDAPEN = 1 (either bank), and there is a write access through an unprivileged DAP request to the DPDLPSTATE register located at address 0xE001E304.

### Implications

If this erratum occurs, an unprivileged debugger write will change the state of the DPLPSTATE register. This can select a lower or higher power mode for the Debug power domain than intended by a privileged debug environment.

Note: Changing the value of the register will not change the functional behavior of the processor. Even if debug domain is powered down, it will be powered up again if a subsequent debug access is made.

### Workaround

There is no workaround provided.

## C00214486

## Floating-point multiple load store instructions which specify unimplemented registers may write to unexpected memory locations and update register incorrectly

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

The VSTMDB/FSTMDBX, VSTMIA/FSTMIAX T1, VLDMDB/FLDMDBX T1 and VLDMIA/FLDMIAX T1 instructions specify a list of floating-point registers to write to/read from memory. The instruction encoding allows a large number of registers to be specified. It has been found that if the specified registers include those not supported by the processor, then data may be written to memory locations above the base address. And the base register or destination register may be updated to wrong value.

### Configurations affected

This erratum affects all configurations of the Cortex®-M52 processor configured with the floating-point or M-profile Vector Extension (MVE) extensions.

### Conditions

- This erratum will occur if the Processing Element (PE) executes a VSTMDB/FSTMDBX instruction with an immediate value greater than 63.
- Or this erratum will occur if the Processing Element (PE) executes a VSTMDB/VSTMIA/VLDMIA/VLDMDB T1 instruction with an immediate value exactly equal to 64 and UInt(D:Vd) is equal to 0.
- Or this erratum will occur if the Processing Element (PE) executes a FSTMDBX/FSTMIAX/ FLDMDBX/ FLDMIAX T1 instruction with an immediate value exactly equal to 65 and UInt(D:Vd) is equal to 0.

### Implications

- The PE will incorrectly write to memory locations immediately above the base address (at most 32 words).
- The PE may incorrectly update base or destination register to an incorrect value.

### Workaround

There is no workaround provided.

## C00215210

## ST_RETIRED PMU event is inaccurate when a CC failed VMRS FPCXT_NS or FPCXT_S instruction executed

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

ST_RETIRED PMU event is wrongly counted when a CC failed VMRS (FPCXT_NS, FPCXT_S) instruction is executed.

### Configurations affected

This erratum affects configurations of the Cortex®-M52 processor configured with Halting debug, DBGLVL > 0.

### Conditions

This erratum is occurred when PMU event counter is programmed to count ST_RETIRED event and PE execute a CC failed VMRS (FPCXT_NS, FPCXT_S) instruction.

### Implications

When above scenario occurs, the PMU event counter will be inaccurate when counting the ST_RETIRED event.

### Workaround

There is no workaround provided.

## C00215763

## Stepping a vector load/store instruction which cause lockup might not enter debug state or DebugMonitor exception when expected

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0. Fixed in r0p1.

### Description

The Armv8-M architecture (in rule RZVKS and rule RMWFT) specifies when Halting debug or DebugMonitor stepping an instruction which causes Lockup, the PE should resolve Lockup signal and enter debug state or DebugMonitor exception if it has sufficient priority. This erratum can cause the stepping vector instruction stay at Lockup but not enter debug state or DebugMonitor exception.

### Configurations affected

This erratum affects all configurations of the Cortex®-M52 processor with the M-profile Vector Extension (MVE) extension implemented (FPMVE =3 or 4 or 5).

### Conditions

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

1. An MVE Vector Load/Store instruction which is subject to beat-wise execution is executed.
2. Beat 0 of the vector load/store instruction has a fault which causes lockup while the beat 1 of the vector instruction stall in DE and be flushed at the same time when stall DE de-asserted.
3. The processor enters lockup as expected but the step event does not occur.

### Implications

When the above scenario happens, the processor enters the lockup state but a debug event does not occur.

- The processor does not Halt or trigger DebugMonitor exception.
- DFSR does not indicate the missed event.

### Workaround

There is no workaround provided.

## C00216080

## PMU event L1D_CACHE is incorrectly implemented in the unified cache

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0 and r0p1. Fixed in r0p2.

### Description

The PMU event L1D_CACHE supported by the Cortex®-M52 processor might be counted incorrectly.

### Configurations affected

This erratum affects the configurations of Cortex®-M52 with unified cache.

### Conditions

A store to the unified cache is followed by a preload instruction.

### Implications

L1D_CACHE

- If it is used to gather statistics on the unified cache, the number of recorded cache accesses will be lower than expected if a store to the unified cache is followed by a preload instruction.

### Workaround

There is no workaround.

## C00216383

## VLDR+VDDUP (with implicit LE) overlap and uncorrectable ECC error occurs, core-ETM gives wrong lockup preferred exception return address

### Status

Affects: Cortex®-M52
Fault type: Programmer category C
Fault status: Present in r0p0, r0p1, fixed in r0p2.

### Description

A MVE vector load/store instruction overlaps with another vector instruction, and the younger one of the overlapped instructions contains an implicit LE, and the MVE vector load/store is executed in the exception handler whose priority is higher than bus-fault. If an uncorrectable ecc fault detected on vector load/store leads core to lockup, and the EPSR.ECI=5, then a wrong address packet would report to trace.

The instruction flow is like this:
0x1100: MVE0
0x1104: MVE1
0x1108: LE <0x1050>

The errata state report preferred exception return address as 0x1104. But based on architecture requirement, it should report address as 0x1050(LE target).

### Configurations affected

This erratum affects the configurations of the Cortex®-M52 processor configured with a DBGLVL>0, ETM=1, ECC=1, FPMVE=3,4,5.

### Conditions

This erratum occurs if ETM function is enabled, and all the following conditions occur:

- A vector load/store instruction overlap with other vector instructions.
- The younger MVE instruction of overlap combines with LE_tag.
- These two instructions execute in the exception handler whose priority is higher than or equal to bus-fault.
- The vector load/store instruction itself triggered an uncorrectable ECC fault, and the ECC fault leads core to lockup.
- The lockup happened with EPSR.ECI=5.

### Implications

When the above scenario happens, the ETM preferred exception return address incorrectly points to the younger instruction of overlap, rather than the LE_tag target address.

### Workaround

There is no workaround.