Arm® Cortex®-M23

Product Revision r0,r1

Software Developers Errata Notice

Non-Confidential - Released

Software Developers Errata Notice

Non-Confidential Proprietary notice

Web Address

http://www.arm.com

Feedback on content

If you have any comments on content, then send an e-mail to errata@arm.com . Give:

- the document title
- the document number, ARM-EPM-107798
- the page numbers to which your comments apply
- a concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Release Information

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 in Chapter 2. Fixed errata are not shown as updated unless the erratum text has changed. The summary table in section 2.2 identifies errata that have been fixed in each product revision.

01 Jul 2019: Changes in Document v6

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

14 Dec 2018: Changes in Document v5

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

15 Nov 2016: Changes in Document v4

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

23 Aug 2016: Changes in Document v3

Page	Status	ID	Cat	Rare	Summary of Erratum
8	New	857327	CatA		TTA instruction in Thread mode might not report correct permission attributes
9	New	857323	CatB		MPU_RLARx.EN bit is not reset on Warm reset
10	Updated	856671	CatC		POP instruction with SP below SPLIM does not behave correctly
13	New	856872	CatC		SCR.SLEEPDEEP can be read from Non-Secure state when SCR.SLEEPDEEPS is set
14	New	856873	CatC		NVIC_ITNSn registers are readable from Non-Secure state
15	New	857022	CatC		A debug write to AIRCR.VECTCLRACTIVE can select the wrong Stack Pointer
16	New	857024	CatC		Setting DHCSR.C_MASKINTS when DHCSR.S_SDE=0 can affect Secure external interrupts
17	New	857321	CatC		Incorrect instruction trace on exit from debug state because of reset
18	New	857324	CatC		Fault on reset entry might enter debug state because of a vector catch
19	New	857326	CatC		Register DAUTHCTRL is RAZ/WI when the debug power domain is off
20	New	857521	CatC		Debug write to AIRCR.VECTCLRACTIVE clears the Secure NMI or Hardfault active bit even if DHCSR.S_SDE is 0

14 Jun 2016: Changes in Document v2

Page	Status	ID	Cat	Rare	Summary of Erratum
7	New	856621	CatA		DIV instruction can update the wrong register when interrupted
10	New	856671	CatC		POP instruction with SP below SPLIM does not behave correctly
11	New	856672	CatC		DCRSR write to CONTROL_S, PRIMASK_S, CONTROL_NS or PRIMASK_NS corrupts R2 or R3 registers
12	New	856675	CatC		A double fault during stacking causes Lockup at the wrong priority level

27 Apr 2016: Changes in Document v1

No new or updated errata in this document version.

See the Summary Table for errata fixes in the latest product revision.

Contents

Chapter 1.
Introduction

This chapter introduces the errata notice for the Cortex-M23 processor.

1.1. Scope of this document

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

- the current status of the defect
- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a 'work-around' where possible

This document describes errata that may impact anyone who is developing software that will run on implementations of this Arm product.

1.2. Categorization of errata

Errata recorded in this document are split into the following levels of severity:

Table 1 Categorization of errata

Errata Type	Definition
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.

Chapter 2.
Errata Descriptions

2.1. Product Revision Status

The r*n*p*n* identifier indicates the revision status of the product described in this book, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

2.2. Revisions Affected

Table 2 below lists the product revisions affected by each erratum. A cell marked with **X** indicates that the erratum affects the revision shown at the top of that column.

This document includes errata that affect revision r0,r1 only.

Refer to the reference material supplied with your product to identify the revision of the IP.

Table 2 Revisions Affected

ID	Cat	Rare	Summary of Erratum	r0p0	r1p0
857327	CatA		TTA instruction in Thread mode might not report correct permission attributes	X	
856621	CatA		UDIV or SDIV instructions can update the wrong register	X	
857323	CatB		MPU_RLARx.EN bit is not reset on Warm reset	X	
857521	CatC		Debug write to AIRCR.VECTCLRACTIVE clears the Secure NMI or Hardfault active bit even if DHCSR.S_SDE is 0	X	
857326	CatC		Register DAUTHCTRL is RAZ/WI when the debug power domain is off	X	
857324	CatC		Fault on reset entry might enter debug state because of a vector catch	X	
857321	CatC		Incorrect instruction trace on exit from debug state because of reset	X	
857024	CatC		Setting DHCSR.C_MASKINTS when DHCSR.S_SDE=0 can affect Secure external interrupts	X	
857022	CatC		A debug write to AIRCR.VECTCLRACTIVE can select the wrong Stack Pointer	X	
856873	CatC		NVIC_ITNSn registers are readable from Non-Secure state	X	
856872	CatC		SCR.SLEEPDEEP can be read from Non-Secure state when SCR.SLEEPDEEPS is set	X	
856675	CatC		A double fault during stacking causes Lockup at the wrong priority level	X	
856672	CatC		DCRSR write to CONTROL_S, PRIMASK_S, CONTROL_NS or PRIMASK_NS corrupts R2 or R3 registers	X	
856671	CatC		POP instruction with SP below SPLIM does not behave correctly	X	

2.3. Category A

856621: UDIV or SDIV instructions can update the wrong register

Category A
Products Affected: Cortex-M23.
Present in: r0p0

Description

A UDIV or SDIV instruction can update the wrong register if interrupted, and access through the Debug Access Port is sent on the AHB bus at the same time.

Configurations affected

This erratum affects configurations with the Debug extensions implemented.

Conditions

1. The processor executes a UDIV or SDIV instruction.

2. The instruction is interrupted on the last cycle.

3. Simultaneously, the debug port accesses the AHB bus and this transfer encounters at least one wait state.

Implications

The address of the destination register is corrupted and any of the registers between R0 and R14 can be written instead of the correct destination register.

Workaround

There is no workaround when a debug connection is active.

857327: TTA instruction in Thread mode might not report correct permission attributes

Category A

Products Affected: Cortex-M23.

Present in: r0p0

Description

When a TTA instruction is executed in Thread mode and Secure state, it uses CONTROL_S.nPRIV instead of CONTROL_NS.nPRIV to check the privilege level and report attributes.

Configurations affected

This erratum affects configurations of the processor with the Security Extension implemented and at least 4 Non-secure MPU regions.

Conditions

1. The CPU runs in Secure state and Thread mode.

2. CONTROL_S.nPRIV is not the same as CONTROL_NS.nPRIV.

3. The CPU executes a TTA instruction to query access permissions of a Non-secure memory location.

Implications

The TTA instruction uses CONTROL_S.nPRIV to report attributes instead of CONTROL_NS.nPRIV. The attributes which are affected are:

- NSRW: Non-secure read and writeable.
- NSR: Non-secure readable.
- RW: Read and writeable.
- R: Readable.

The attributes are reported as:

- When CONTROL_S.nPRIV=1 and CONTROL_NS.nPRIV=0, fields NSRW, NSR, R, and W report 0 but access might be permitted in Non-secure state.
- When CONTROL_s.nPRIV=0 and CONTROL_NS.nPRIV=1, fields NSRW, NSR, R, and W might report Non-Zero values but access might not be permitted in Non-secure state.

Workaround

There is no workaround.

2.4. Category A (Rare)

There are no errata in this category

2.5. Category B

857323: MPU_RLARx.EN bit is not reset on Warm reset

Category B
Products Affected: Cortex-M23.
Present in: r0p0

Description

The region enable bit MPU_RLARx.EN is not reset on a Warm reset in the RAR=0 configuration.

Configurations affected

This erratum affects configurations of the processor where RAR=0.

Conditions

For any of the MPUs, Secure and/or Non-secure:

1. Some MPU regions are not programmed after a Warm reset: MPU_RLAR registers are not all written.

2. The MPU is enabled: MPU_CTRL.ENABLE is set to 1.

Implications

Registers MPU_RLARx are not reset on a Warm reset. The enable bit might still be set to 1 after the reset and cause the region to be enabled. This can cause any of:

- Access to unprivileged code to a memory region that has not been enabled.
- Code execution to a memory region that has not been permitted.
- A fault because of a hit in two regions at the same time.

Workaround

Before enabling the Secure or Non-secure MPU, the MPU_RLAR register of each region should be written. For the regions that are not used, bit MPU_RLAR.EN must be set to 0.

2.6. Category B (Rare)

There are no errata in this category.

2.7. Category C

856671: POP instruction with SP below SPLIM does not behave correctly

Category C
Products Affected: Cortex-M23.
Present in: r0p0

Description

A POP instruction for which the Stack Pointer is below the SPLIM register before execution and is greater than or equal to the SPLIM register after execution might not correctly report the stack limit fault, or read an address below the SPLIM register.

Configurations affected

This erratum affects configurations with the Security Extension implemented.

Conditions

1. The processor runs in Secure state.

2. One SPLIM register (PSPLIM or MSPLIM) corresponding to the current Stack Pointer (PSP or MSP) is programmed to a non-zero value.

3. The current Stack Pointer is at a value lower than the corresponding SPLIM register.

4. A POP instruction loading 'numregs' registers is executed, so that SP + 4 * 'numregs' is higher than or same as the corresponding SPLIM.

An additional condition changes the implications:

5. The initial address of the Stack Pointer is SPLIM-0x4.

Implications

If conditions 1. to 4. are met, then:

- The first read access of the POP instruction is performed whereas it is below the SPLIM value.

If conditions 1. to 5. are met, then:

- The first read access of the POP instruction is performed whereas it is below the SPLIM value, and the fault is not reported.

Workaround

No workaround is required in standard conditions. The Stack Pointer should never be below the SPLIM register. If an instruction tries to set the Stack Pointer below the SPLIM register, then a fault is detected and the Stack Pointer is either:

- Not modified, for example on a PUSH, ADD, SUB, or MOV instruction, or
- Changed to the SPLIM value on exception entry.

To ensure that this condition is met, you must ensure that:

- When writing to the Stack Pointer with an MSR instruction, the final value is not below the SPLIM register.
- When modifying the SPLIM register with a MSR instruction, the value of the SPLIM register is below the corresponding Stack Pointer.

856672: DCRSR write to CONTROL_S, PRIMASK_S, CONTROL_NS or PRIMASK_NS corrupts R2 or R3 registers

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

A debug write to the DCRSR register to update the PRIMASK_S or CONTROL_S registers corrupts the R2 register. A debug write to the DCRSR register to update the PRIMASK_NS or CONTROL_NS registers corrupts the R3 register.

Configurations affected

This erratum affects configurations with both the Security and Debug Extensions implemented.

Conditions

1. The processor is in Halt state with Secure Intrusive debug enabled.

2. The debugger writes to the DCRSR register to update one of PRIMASK_S, PRIMASK_NS, CONTROL_S, or CONTROL_NS.

Implications

The requested register is modified correctly, but at the same time:

- Register R2 is corrupted if CONTROL_S or PRIMASK_S is selected.
- Register R3 is corrupted if CONTROL_NS or PRIMASK_NS is selected.

Workaround

Registers R2 and R3 need to be saved and restored:

- Saved when entering Halt mode and restored when leaving Halt mode, or
- Saved before updating one of CONTROL_S, PRIMASK_S, CONTROL_NS, or PRIMASK_NS, and restored after the update.

856675: A double fault during stacking causes Lockup at the wrong priority level

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

When a fault is detected during NMI stacking and BFHFNMINS is set to 1, another fault might be detected and cause Lockup at the wrong priority level (-3 instead of -2).

Configurations affected

This erratum affects configurations with the Security Extension implemented.

Conditions

1. The processor runs in Secure state.

2. AIRCR.BFHFNMINS is set to 1.

3. Hardfault Non-secure (priority -1) is active.

4. An NMI (priority -2) preempts the CPU, which starts a stacking phase.

5. During the last access of the Caller registers stacking (XPSR register), a bus fault is detected.

6. The CPU detects a Secure fault during the first address phase of the Callee registers (R4 register). This fault can be an MPU or SPLIM fault.

Implications

The CPU enters Lockup because Non-secure Hardfault is active. However, the Lockup should happen at the NMI priority (-2, Non-secure), but is at the Secure Hardfault priority (-3).

It means that a debugger can make the CPU leave Lockup state only if Secure Invasive Debug is authenticated, but Non-secure Invasive Debug should have been enough to exit Lockup state.

Workaround

No workaround is required in normal conditions as Arm recommends leaving Lockup state by resetting the CPU instead of using a debugger.

Authenticating Secure Invasive Debug allows the processor to exit Lockup state.

856872: SCR.SLEEPDEEP can be read from Non-Secure state when SCR.SLEEPDEEPS is set

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

SCR.SLEEPDEEP is defined as RAZ/WI from Non-Secure state when SCR.SLEEPDEEPS is set. In the Grebe processor, this bit is WI but reads the value.

Configurations affected

This erratum affects configurations with the Security Extension implemented.

Conditions

1. SCR.SLEEPDEEP is set to 1.

2. SCR.SLEEPDEEPS is set.

Implications

If the processor reads the value of SCR.SLEEPDEEP from Non-Secure state, the register bit will read as 1, but this register bit is defined as RAZ/WI in the architecture.

Workaround

No workaround is required. This erratum provides more information than expected to Non-Secure code, but this information is not useful.

856873: NVIC_ITNSn registers are readable from Non-Secure state

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

The NVIC_ITNSn registers are defined as RAZ/WI from Non-Secure state. Because of this erratum, the processor can read the value of these registers from Non-Secure state.

Configurations affected

This erratum affects configurations with the Security Extension implemented.

Conditions

1. The value of the NVIC_ITNSn registers contains bits that read as 1 for interrupts which are present in the design.

Implications

If the processor reads the NVIC_ITNSn registers, it is able to read the interrupts which target Non-Secure state but it is not able to modify these registers.

Workaround

No workaround is required as no useful information can be read from these registers. These registers list the interrupts that target Non-Secure state, but Non-Secure state can get the same information by writing to the NVIC_ISERn registers and checking which bits can be set to 1. Only the interrupts for which the corresponding bit in the NVIC_ITNSn registers is set can be enabled from Non-Secure state.

857022: A debug write to AIRCR.VECTCLRACTIVE can select the wrong Stack Pointer

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

When the CPU is in debug state, a debugger can use bit VECTCLRACTIVE of register AIRCR to clear active state of interrupts and set IPSR to 0. Until the CPU leaves debug state, the selected Stack Pointer register might not be correct.

Configurations affected

This erratum affects configurations of the processor with the Debug extensions implemented.

Conditions

1. The CPU is halted in debug state.

2. The CPU is in Non-secure state, or DHCSR.S_SDE is set.

3. The CPU is in Handler mode: IPSR is Non-Zero and at least one exception is active.

4. CONTROL.SPSEL is set for the current Security state.

Implications

IPSR is cleared when AIRCR.VECTCLRACTIVE is set to 1 and the CPU changes to Thread mode. If CONTROL.SPSEL is set for the current Security state, the current Stack Pointer should point to the PSP register, but the MSP register will continue to be used.

The Stack Pointer will point to PSP until the CPU leaves debug state.

Workaround

A workaround is only required when the CPU is halted in debug state after AIRCR.VECTCLRACTIVE is set to 1. Until the CPU leaves debug state, a debugger should explicitly read or update SP_main or SP_process instead of the current Stack Pointer.

For example, instead of writing this value to DCRSR.REGSEL:

- 0b0001101 current Stack Pointer, SP.

Use one of these values depending on the CONTROL.SPSEL register value:

- 0b0010001 current state main Stack Pointer, SP_main.
- 0b0010010 current state process Stack Pointer, SP_main.

857024: Setting DHCSR.C_MASKINTS when DHCSR.S_SDE=0 can affect Secure external interrupts

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

Register DHCSR should only affect Non-secure interrupts when DHCSR.S_SDE=0, but revision r0p0 of Grebe masks all external interrupts (Secure or Non-secure) if the interrupt with the highest priority is Non-secure.

Configurations affected

This erratum affects configurations of the processor with the Debug and Security Extensions implemented.

Conditions

1. DHCSR.S_SDE is set to 0.

2. The external interrupt with the highest priority is Non-secure and there is no other exception with a higher priority.

Implications

C_MASKINTS masks the pending external interrupt with the highest priority when it is Non-secure, even if a Secure interrupt could preempt if the Non-secure one was masked independently.

SysTick and PendSV are not impacted because they are masked independently.

Workaround

If Non-secure debug is permitted, the Secure application needs to be aware that Secure external interrupts might not be able to preempt when DHCSR.C_MASKINTS is set.

857321: Incorrect instruction trace on exit from debug state because of reset

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

When the CPU is halted in debug state and a reset occurs, the ETM trace contains a Reset Exception Entry packet and an I-Sync packet, but only the I-Sync packet should be sent.

Configurations affected

This erratum affects configurations of the processor with the Debug extensions implemented.

Conditions

1. The CPU is halted in debug state.

2. The ETM trace is enabled.

3. The system or debugger asserts system reset.

Implications

Because of this erratum, the trace contains the following packets:

1. A Reset Exception Entry packet with address 0x00000000.

2. An I-Sync packet (exit from debug state) with the address of the reset handler.

Only the second packet should be sent.

Workaround

The debugger should ignore the first packet.

857324: Fault on reset entry might enter debug state because of a vector catch

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

A debugger can set DEMCR.VC_HARDERR to halt to debug state when a Hardfault exception occurs. When the CPU resets on a system reset and gets a fault during the stack or vector fetch, it might halt to debug state instead of entering lockup state.

Configurations affected

This erratum affects configurations of the processor with the Debug extensions implemented.

Conditions

1. Intrusive debug is enabled.

2. If the CPU implements the Security Extension, DHCSR.S_SDE is 1.

3. DEMCR.VC_HARDERR is set to enter debug state on a Hardfault exception.

4. A system reset resets the CPU.

5. Access to MSP value or reset handler address returns a bus fault.

Implications

The CPU should enter lockup state with IPSR value set to 0. Because it is detected as a Hardfault exception, it will instead halt to debug state.

Workaround

The debugger should check the value of IPSR and PC value to detect this case.

857326: Register DAUTHCTRL is RAZ/WI when the debug power domain is off

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

The CPU can read or write register DAUTHCTRL to override Secure debug authentication. When the debug power domain is off, this register becomes RAZ/WI and is reset on a debug reset.

Configurations affected

This erratum affects configurations of the processor with the Security and Debug Extensions implemented.

Conditions

1. The debug power domain is turned off.

2. The CPU tries to access register DAUTHCTRL.

Implications

Register DAUTHCTRL becomes RAZ/WI. As long as the debug power domain is off, this has no impact. However, it is not possible to write a value until a debugger is connected and the debug power domain is turned on.

This register is also reset on a debug reset.

Workaround

Because of this erratum, register DAUTHCTRL should only be written when a debugger is connected, or when the debug power domain is turned on.

857521: Debug write to AIRCR.VECTCLRACTIVE clears the Secure NMI or Hardfault active bit even if DHCSR.S_SDE is 0

Category C

Products Affected: Cortex-M23.

Present in: r0p0

Description

If the Secure handler calls a Non-secure function and the CPU is halted in Debug state, a Debug write to AIRCR.VECTCLRACTIVE might clear the Secure NMI or Hardfault active bit.

Configurations affected

This erratum affects configurations of the processor with both the Debug and Security extensions implemented.

Conditions

1. The CPU runs the Secure NMI or Hardfault handler.

2. The handler calls a Non-secure function before returning from an exception.

3. Debug is enabled and the CPU is halted.

4. The debugger sets the AIRCR.VECTCLRACTIVE bit.

Implications

A Debug write to AIRCR.VECTCLRACTIVE can clear the current active exception without clearing IPSR. Only NMI and Hardfault exceptions are impacted because they have a negative priority.

If the CPU runs the Secure Hardfault or NMI handler and calls a Non-secure function, the Non-secure code can clear the active bit of the highest priority exception with a negative priority. However, in this case, a fault is triggered.

Workaround

The CPU that runs the Secure NMI or Hardfault handler should never call a Non-secure function before returning from an exception.

If the NMI or Hardfault exceptions must be handled by Non-secure code, you can enable this functionality by setting the AIRCR.BFHFNMINS bit.