Arm® CoreLink™ SSE-200 Subsystem for Embedded **Software Developer Errata Notice**

This document contains all known errata since the r1p0 release of the product.

Non-Confidential Proprietary notice

LES-PRE-20349

Confidentiality Status

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

Web address

http://www.arm.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 then send an e-mail to errata@arm.com giving:

- The document title.
- The document number: SDEN-956540.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm 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 7 identifies errata that have been fixed in each product revision.

25-Jan-2019: Changes in document version 5.0				
ID	Status	Area	Cat	Summary of erratum
1332414	New	Programmer	CatB	Permanently secure peripherals can be configured as non-secure

13-Jul-2018: Changes in document version 4.0				
ID	Status	Area	Cat	Summary of erratum
1180706	New	Programmer	CatC	IRQs sources of peripherals located in PD_SYS power domain can not prevent PD_SYS domain entering power OFF state
1159980	New	Programmer	CatC	Missing definition of EWC wake up capabilities for internal IRQs
1137419	New	Programmer	CatC	EWC and WIC do not pend all pulse interrupts
1124551	New	Programmer	CatC	Un-trusted debugger may power up CPUs

29-Mar-2018: Changes in document version 3.0				
ID	Status	Area	Cat	Summary of erratum
977925	New	Programmer	CatC	Reset syndrome register does not reflect the actual reset source in some conditions

15-Jan-2018: Changes in document version 2.0				
ID	Status	Area	Cat	Summary of erratum
1009975	New	Programmer	CatA	EWC load not having effect
1002571	New	Programmer	CatB	SRAM macro clock toggle by isolation
974698	Updated	Programmer	CatB	CPU stuck in warmreset during EWC based wake up from OFF state

29-Sep-2017: Changes in document version 1.0				
ID	Status	Area	Cat	Summary of erratum
974698	New	Programmer	CatB	CPU stuck in warmreset during EWC based wake up from OFF state

Errata summary table

The errata associated with this product affect product versions as below.

ID	Cat	Summary	Found in versions	Fixed in version
1332414	CatB	Permanently secure peripherals can be configured as non-secure	r1p0, r2p0	open
1180706	CatC	IRQs sources of peripherals located in PD_SYS power domain can not prevent PD_SYS domain entering power OFF state	r1p0, r2p0	open
1159980	CatC	Missing definition of EWC wake up capabilities for internal IRQs	r1p0	r2p0
1137419	CatC	EWC and WIC do not pend all pulse interrupts	r1p0, r2p0	open
1124551	CatC	Un-trusted debugger may power up CPUs	r1p0, r2p0	open
1009975	CatA	EWC load not having effect	r1p0	r2p0
1002571	CatB	SRAM macro clock toggle by isolation	r1p0	r2p0
977925	CatC	Reset syndrome register does not reflect the actual reset source in some conditions	r1p0	r2p0
974698	CatB	CPU stuck in warmreset during EWC based wake up from OFF state	r1p0	r2p0

Errata descriptions

Category A

1009975

EWC load not having effect

Status

Affects: CoreLink SSE-200
Fault type: Programmer Category A
Fault status: Present in r1p0. Fixed in r2p0

Description

After a clear request, if the new set request has been asserted during first External Wakeup Controller (EWC) arming, the EWC is not activated again.

Conditions

- EWC is enabled.
- Processor goes to Wait For Interrupt (WFI), EWC set requested.
- EWC arming starts.
- IRQ asserted during EWC arming, EWC clear requested.
- IRQ served during EWC arming.
- Processor goes to WFI, new EWC set requested.
- New EWC set request is cleared when EWC arming is finished.
- EWC disarming finishes but arming is not started.
- Processor goes to OFF without active EWC.

Implications

Deadlock: Processor is OFF instead of OFF with EWC, IRQ and Non-Maskable Interrupt (NMI) cannot wake up the processor.

Workaround

No workaround.

Category A (rare)

There are no errata in this category.

Category B

1332414

Permanently secure peripherals can be configured as non-secure

Status

Affects: CoreLink SSE-200
Fault type: Programmer Category B

Fault status: Present in r1p0,r2p0

Description

The following peripherals of SSE-200 follow the security configuration of the S32KTIMER.
However, all of these peripherals are expected to be permanently secure.

- S_SYSCONTROL,
- SYS_PPU,
- CPU0CORE_PPU,
- CPU0DBG_PPU,
- CPU1CORE_PPU,
- CPU1DBG_PPU,
- CRYPTO_PPU,
- DBG_PPU,
- RAM0_PPU,
- RAM1_PPU,
- RAM2_PPU,
- RAM3_PPU,
- S32KWATCHDOG

Configurations affected

All configurations of the CoreLink SSE-200.

Conditions

The erratum arises when the following condition is true:
APBNSPPC1.NS_S32K bit is set to 1'b1.

Implications

All the following implications hold:

- Secure accesses towards Secure peripherals in the 0x5002_0000-0x5003_FFFF region may generate all the following:
    - Security violation interrupt, if SECPPCINTEN.S_APBPPC1PERIP_EN ==1'b1
    - Bus error, if SECRESPCFG.SECRESPCFG == 1'b1
- The following peripherals are unavailable to any memory accesses:
    - S_SYSCONTROL,
    - SYS_PPU,
    - CPU0CORE_PPU,
    - CPU0DBG_PPU,
    - CPU1CORE_PPU,
    - CPU1DBG_PPU,
    - CRYPTO_PPU,
    - DBG_PPU,
    - RAM0_PPU,
    - RAM1_PPU,
    - RAM2_PPU,
    - RAM3_PPU,
    - S32KWATCHDOG

Workaround

The issue can be mitigated by means of the following workaround:

Software workaround

- Non-secure software awareness is required:
    - Permanently keep the S32K Timer secure by configuring APBNSPPC1.NS_S32K == 0
        - Implement a secure API that is non-secure callable, and that the non-secure world can use to access the secure timer. The corresponding timer interrupt can remain non-secure, and the non-secure handler can use the timer through this secure API.
        - Do not use the timer from non-secure mode.
- Transparent to non-secure software:
    - The secure world must temporarily switch the 32K Timer back to secure when accessing the affected secure peripherals, and make sure that the non-secure IRQ handler cannot interrupt during this period. This may require the following:
        - A secure NMI handler, achieved by configuring AIRCR.BFHFNMINS == 0.
        - Prioritization of secure exceptions, achieved by configuring AIRCR.PRIS == 1.

Hardware workaround

No hardware workaround is available.

1002571

SRAM macro clock toggle by isolation

Status

Affects: CoreLink SSE-200
Fault type: Programmer Category B
Fault status: Present in r1p0. Fixed in r2p0

Description

There is a race condition between the Power Control State Machine (PCSM) in the Base element and the PCSM in the SRAM element.

Conditions

The race condition exists if the clock force is asserted before Warm reset, which might occur as part of many power state transitioning scenarios. Here are a few examples:

- After Warm reset:
  If SRAM PCSM is faster than the Base PCSM, that is, the SRAM clocks are enabled earlier, then the Base isolation is removed.
- If SRAM wake-up is triggered by the PPU while the processor is in a Wake For Interrupt (WFI) state:
  If SRAM PCSM is slower than the Base PCSM, then Base goes off while SRAM comes on.
  When a wake-up event is sent to the Base PPU, the SRAM clock isolation is removed while the SRAM clock is running already.

It is not required to list all the conditions, since the workaround resolves the root cause in all conditions.

Implications

The SRAM clock might suffer from pulse width violation while it is ON but not selected. The entailed consequences are not determined.

Workaround

Avoid using the SRAM clock force (the SRAMSYSCLK_FORCE bit in the CLOCK_FORCE register). It is cleared after a Cold reset.

974698

CPU stuck in warmreset during EWC based wake up from OFF state

Status

Affects: CoreLink SSE-200
Fault type: Programmer Category B
Fault status: Present in r1p0. Fixed in r2p0

Description

This is a race condition between the warmreset procedure and a wakeup event.

Conditions

One of the CPUs is in External Wakeup Controller based OFF state (EWCCTRL.EWCxEN_STATUS == 1) , the other CPU is in ON state.
From this state when a warm-reset is requested using the AIRCR.SYSRESETREQ register of the CPU and in parallel a wakeup event triggers the other CPU to turn on.

Implications

There is a potential of the woken up CPU to get stuck in warm reset state.

Workaround

- Under normal operation conditions, it is not recommended for software to use the AIRCR.SYSRESETREQ to reset the system. This type of reset can be disabled using RESET_MASK.SYSRSTREQx_EN register fields. it is expected that software uses SWRESET.SWRESETREQ register field to reboot the system.
- Under debug operation conditions, this bug can be avoided if the debugger use nSRST reset input to reset the system. Note you lose the debug state with this method, you can consider using the CPUWAIT registers to prevent CPUs code execution after reset to be able to restore debug state.
- This lock situation is rare and the condition can be identified by the debugger and initiate reset via nSRST or SWRESET.SWRESETREQ registr field. The sequence to identify the condition is (using CPU0 for warm-reset initiation):
    - Initiate warm-reset via AIRCR.SYSRESETREQ
    - Wait for CPU0 to come out of warm-reset (using CPU0.DHCSR.S_RESET_ST)
    - Read out CPU1 PPU PPU_PWSR.PWR_STATUS
    - If the status was 4'b1001 (WRST) then the CPU1 got stuck in warm reset state
    - Initiate full system reset using nSRST or writing to SWRESET.SWRESETREQ
    - Note you lose the debug state with this method. If you used nSRST you can consider using the CPUWAIT registers to prevent CPUs code execution after reset to be able to restore debug state.

Category B (rare)

There are no errata in this category.

Category C

1180706

IRQs sources of peripherals located in PD_SYS power domain can not prevent PD_SYS domain entering power OFF state

Status

Affects: CoreLink SSE-200
Fault type: Programmer CatC
Fault status: Present in r1p0, r2p0

Description

The following IRQ sources of peripherals that are located in the PD_SYS power domain can not prevent the PD_SYS power domain from entering power OFF state:

- IRQ[0] NON-SECURE WATCHDOG Reset Request
- IRQ[1] NON-SECURE WATCHDOG Interrupt
- ~~IRQ[2] S32KTimer~~ (not expected to keep up PD_SYS)
- IRQ[3] TIMER 0
- IRQ[4] TIMER 1
- IRQ[5] DUAL TIMER
- IRQ[6] Message Handling Unit 0 CPUn Interrupt
- IRQ[7] Message Handling Unit 1 CPUn Interrupt
- ~~IRQ[8] CryptoCell 312 (if Crypto Element is present)~~ (not expected to keep up PD_SYS)
- IRQ[9] MPC Combined (Secure)
- IRQ[10] PPC Combined (Secure)
- IRQ[11] MSC Combined (Secure)
- IRQ[12] Bridge Error Combined Interrupt (Secure)

Configurations affected

All configurations

Conditions

The PD_SYS power domain enters the OFF state when the following conditions are satisfied:

- Both of the CPUs are in either of the following states:
  - "OFF – DeepSleep with WIC + EWC"
  - "OFF"
- The Power Policy of the PD_SYS PPU is set to Dynamic OFF (for more information see ARM DEN 0051E):
  - PPU_PWPR.PWR_DYN_EN = '1'
  - PPU_PWPR.PWR_POLICY= '0000'
- No Active keep up conditions for PD_SYS are present, taking in to account the PDCM_PD_SYS_SENSE memory mapped register.
- The hardware status enables the PD_SYS power domain to enter low-power mode (both internal and expansion interface Q-Chanel SYSPWRQ* can enter quiescent state).

Implications

- These IRQs are lost if PD_SYS enters OFF state, that is, the system enters HIBERNATION state.

Workaround

Software workaround

These workarounds are expected to remain functional in case of a future hardware fix:

- If PDCM_PD_SYS_SENSE.S_PD_SYS_ON == 1, there is no IRQ loss but it prevents the system going into HIBERNATION low power state.
- Software has to disable these IRQ sources and ensure that the software does not depend on these IRQs before entering HIBERNATION state.

Hardware workaround

There is no hardware workaround.

1159980

Missing definition of EWC wake up capabilities for internal IRQs

Status

Affects: CoreLink SSE-200
Fault type: Programmer CatC
Fault status: Present in r1p0. Fixed in r2p0

Description

None of the SSE-200 internal interrupt sources, except NMI (S32KWDog) and IRQ[2] (S32KTimer), are EWC wake-up capable.
The update of the architecture specification now defines the following additional IRQs as EWC wake-up capable:

- IRQ[0] NON-SECURE WATCHDOG Reset Request
- IRQ[1] NON-SECURE WATCHDOG Interrupt
- IRQ[3] TIMER 0
- IRQ[4] TIMER 1
- IRQ[5] DUAL TIMER
- IRQ[6] Message Handling Unit 0 CPUn Interrupt
- IRQ[7] Message Handling Unit 1 CPUn Interrupt
- IRQ[8] CryptoCell-312 (if Crypto Element is present)
- IRQ[9] MPC Combined (Secure)
- IRQ[10] PPC Combined (Secure)
- IRQ[11] MSC Combined (Secure)
- IRQ[12] Bridge Error Combined Interrupt (Secure)

Configurations affected

All configurations

Conditions

When both the following conditions are true for either or both CPUs:

- The CPU is in the "OFF – DeepSleep with WIC + EWC" state.
- IRQ[0-1] and IRQ[3-12] are enabled in the NVIC and not masked by priority.

Implications

1. None of the above internal interrupt sources can wake up the CPU from EWC OFF.
2. These IRQs are lost if PD_SYS goes to OFF, with the exception of IRQ[8] CryptoCell-312 (if Crypto Element is present).

Workaround

The following workarounds are provided for the corresponding implications:

Software workaround

1. There is no software workaround.
2. If PDCM_PD_SYS_SENSE.S_PD_SYS_ON == 1, there is no IRQ loss but it prevents the system entering HIBERNATION state. This workaround is expected to remain functional in case of a future hardware fix.

Hardware workaround

No hardware workaround is provided.

1137419

EWC and WIC do not pend all pulse interrupts

Status

Affects: CoreLink SSE-200
Fault type: Programmer CatC
Fault status: Present in r1p0, r2p0

Description

There is a functional mismatch between the External Wakeup Controller (EWC) and the WIC, compared with the CPU core's NVIC, in pulse interrupt cases.

- When in OFF or WIC sleep low power state, both the EWC and the WIC guarantee to pend properly configured (during SoC integration) pulse interrupts only.
- Conversely, in NVIC Sleep power state, the CPU core's NVIC pends *all* implemented interrupts.

Note:
Pended interrupts do not necessary wake up the CPU core and cannot be active until they are enabled in the NVIC, and their execution priority is below the interrupt priority.
Wake up sensitivity is defined by the current wake up mask. You must be aware that handler mode might manipulate the interrupt/wake up mask value based on the current execution priority level. For more information see the Arm® v8-M Architecture Reference Manual lDDI0553A.

Configurations affected

If CPU<n> interrupt pin <x> has the following parameters set, then pulse interrupts are not captured by the EWC and the WIC:
If the interrupt is NMI:

- CPU<n>_EXP_NMI_PULSE_SPT_EN == ‘0’.

For all other interrupts...:

- CPU<n>_EXP_IRQ_PULSE_SPT_EN<x> == ‘0’.

...and the corresponding interrupt is implemented:

- CPU<n>_EXP_IRQDIS<x> == '0'
- CPU<n>_EXP_NUMIRQ >= <x>

Conditions

The condition arises when a CPU enters one of the following low power modes:

- "OFF – DeepSleep with WIC + EWC"
- "RET – DeepSleep with WIC"
- "ON - DeepSleep with WIC"

Implications

Improperly configured pulse interrupts are not pended during these sleep modes, compared with the "ON - Sleep" power mode which always pends all implemented interrupts.

Workaround

An improved description has been added to programmers model in Technical Reference Manual. No hardware fix is provided as this is not a bug.

Software workaround

No software workaround is provided.

Hardware workaround

No hardware workaround is provided.

1124551

Un-trusted debugger may power up CPUs

Status

Affects: CoreLink SSE-200
Fault type: Programmer CatC
Fault status: Present in r1p0.r2p0

Description

Architecturally, the Granular Power Requestors (GPR) in the debug infrastructure, are accessible and controllable by an external debugger, irrespective of debug authentication status. If either, or both, of the CPUs are in the "OFF" or "OFF – DeepSleep with WIC + EWC" mode, and do not have a valid vector table in memory along with a corresponding vector table offset, un-trusted parties can trigger the power up of either, or both, CPU cores, which starts execution in secure mode from a possibly invalid reset vector location with an invalid stack pointer.

Configurations affected

All configurations

Conditions

If CPU<x> is in either of the following modes:

- "OFF"
- "OFF – DeepSleep with WIC + EWC"

and does not have a valid vector table in memory, along with a corresponding vector table offset that is defined in:

- INITSVTOR<x>.INITSVTOR<x>

Implications

An un-trusted debugger can power up either CPU core, which then starts executing from an unintended location in secure mode, using an unintended stack pointer.

Workaround

The following software workaround has been added to the programmers model in the Technical Reference Manual. There is no hardware fix provided.

Software workaround

To prevent CPU<x> running invalid code, at least one of the following is needed:

- Ensure that boot vector INITSVTOR<x> is pointing to valid boot code locations, even if it is simply to place the CPU back to DeepSleep and WFI, and therefore to turn off the CPU. This ensures that no invalid code will ever be executed.
- Set the associated CPUWAIT bits of the CPU<x> core that is not expected to run, to HIGH, or to default to HIGH at reset. This ensures that CPU<x> does not run code or can not power up without secure software or secure debug clearing the associated CPUWAIT bits.

Hardware workaround

No hardware workaround is provided.

977925

Reset syndrome register does not reflect the actual reset source in some conditions

Status

Affects: CoreLink SSE-200
Fault Type: Programmer Category C
Fault status: Present in r1p0. Fixed in r2p0

Description

The reset syndrome register does not reflect the actual reset source, in some conditions.

Configurations affected

This erratum affects all configurations of the CoreLink SSE-200.

Conditions

Condition 1

- Warm reset sequence initiated by AIRCR.SYSRESETREQ register of the CPU(s)
- Cold Reset request occurs (SWRESET, Secure/Non-secure Watchdog, EXTRESETREQ)
- Corresponding RESET_SYNDROME register bit is set of the Cold Reset
- Warm reset sequence finished
- RESET_SYNDROME.SYSRESETREQ<x> is set
- Software reads RESET_SYNDROME
- It seems both Warm reset and Cold reset occurred
- Software clears RESET_SYNDROME
- Cold reset occurs
- After reset, software reads RESET_SYNDROME and it is empty

Condition 2

- Cold Reset request occurs (SWRESET, Secure/Non-secure Watchdog, EXTRESETREQ)
- Corresponding RESET_SYNDROME register bit is set of the Cold Reset
- Software reads RESET_SYNDROME
- It seems the Cold reset occurred
- Software clears RESET_SYNDROME
- Cold reset occurs
- After reset, software reads RESET_SYNDROME and it is empty

Implications

- Software will not be able to determine the reset source after Cold reset.

Workaround

- If software sees empty RESET_SYNDROME after reset, it should assume that there was a Cold reset (nSRST, SWRESET, EXTRESET, WatchDog reset).