TRAVEO™ T2G CYT2BL errata sheet

Marking/Step: Rev. A

32-bit single-chip microcontroller

About this document

Scope and purpose

This document describes the deviations of the device from the current user documentation, to support the assessment of the effects of these deviations on your custom hardware and software implementations.

Please take note of the following information:

- This errata sheet applies to all temperature and frequency versions and to all memory size variants, unless explicitly noted otherwise. For a derivative synopsis, see the latest datasheet or user manual
- Multiple device variants are covered in this one document. If an issue is related to a particular module, and this module is not specified for a specific device variant, then the issue does not apply to that device variant
- Devices marked with ES are engineering samples which may not be completely tested in all functional and electrical characteristics and are therefore only suitable for evaluation
- Some of the errata have workarounds which may be supported by the tool vendors. Some corresponding compiler switches may need to be set. Please refer to the respective documentation of your compiler
- To understand the effect of issues relating to the on-chip debug system, please refer to the respective debug tool vendor documentation

Current documentation

- TRAVEO™ T2G automotive MCU body controller entry architecture technical reference manual
- TRAVEO™ T2G automotive MCU body controller entry registers technical reference manual for CYT2BL
- CYT2BL datasheet TRAVEO™ T2G automotive MCU

Newer versions replace older versions, unless specifically stated otherwise.

Please always refer to the corresponding documentation for this device available in the category 'Documents' at www.infineon.com/TRAVEO™ and www.myInfineon.com.

Conventions used in this document

Each erratum identifier follows the pattern [Number]:

- [Number] = ascending sequential number within the three

  ***Note**:* *[Number] As this sequence is used over several derivatives, including already solved deviations, gaps can occur inside this numbering sequence*

Table of contents

About this document . . . 1
Table of contents . . . 2
1 Errata overview . . . 3
2 Functional deviations . . . 5
Revision history of CYT2BL errata sheet . . . 35
Disclaimer . . . 36

1 Errata overview

List of errata referenced in this document.

Table 1 Functional deviations

Issue title	Change	Page
[067]ConfigureFmInterrupt API assumes a parameter with 8 bytes boundary, but actual boundary is 4 bytes		5
[068]SMPU/MPU/PPU protection region size is limited to 2 GB		5
[069]DirectExecute API may return error if called with arguments placed in SRAM memory		6
[096] CAN FD RX FIFO top pointer feature does not function as expected		6
[097] CAN FD debug message handling state machine not get reset to Idle state when CANFD_CH_CCCR.INIT is set		7
[098]TPIU peripheral ID mismatch		7
[137]Limitation of work flash reading		8
[138]ROM boot code clears to zero last 2 KB of SRAM		9
[139]Limitation of programming SFlash Normal Access Restrictions (row 13)		9
[147]CAN FD controller message order inversion when transmitting from dedicated Tx Buffers configured with same Message ID		10
[162]Temperature sensor accuracy issue		11
[167]CAN FD incomplete description of Dedicated Tx Buffers and Tx Queue related to transmission from multiple buffers configured with the same Message ID		11
[175]Misleading status is returned for Flash and eFuse system calls if there are pending NC ECC faults in SRAM controller #0		12
[176]WDT reset causes loss of SRAM retention		13
[185]Crypto ECC errors may be set after boot with application authentication		13
[198]Incomplete erase of Code Flash cells could happen Erase Suspend / Erase Resume is used along with Erase Sector operation in Non-Blocking mode		14
[199]Limitation for keeping the port state from peripheral IP after wakeup from DeepSleep		15
[201]A part of the PWR_CTL2.BGREF_LPMODE description is lacked in the existing register TRM		16
[202]Limitation of clock configuration before entering DeepSleep mode		16
[203]Several data retention information in Register TRM are incorrect		17
[204]SCBx_INTR_TX.UNDERFLOW bit may be set unintentionally		18
[206] Hardfault may occur when the SROM APIs listed below while executing EraseSector or ProgramRow in non-blocking mode		18
[209] CAN FD sporadic data corruption (payload) in case acceptance filtering is not finished before reception of data R3 (DB7..DB4) is completed		19

(table continues...)

1 Errata overview

Table 1 (continued) Functional deviations

Issue title	Change	Page
[212] Description for PASS SARx to TCPWMx direct connect triggers one-to-one is incorrect in datasheet		23
[218]Greater change of Low voltage detection (LVD) level can issue an Over voltage detection (OVD) reset		24
[229]System calls: improper usage of memory region end address in control logic	changed	25
[242]RTC busy status bit stuck issue	new	26
[740]Interrupted loads to SP can cause erroneous behavior (Arm® CM4 errata ID: 752770)	new	28
[741]VDIV or VSQRT instructions might not complete correctly when very short ISRs are used (Arm® CM4 errata ID: 776924)	new	29
[742]ITM can deadlock when global timestamping enabled (Arm® CM4 errata ID: 806422)	new	30
[743]Store immediate overlapping exception return operation might vector to incorrect interrupt (Arm® CM4 errata ID: 838869)	new	31
[744]Fused MAC instructions give incorrect results for rare data combinations (Arm® CM4 errata ID: 839676)	new	32
[745]Processor reset asserted asynchronously could corrupt FPB comparator registers and remap to wrong address (Arm® CM4 errata ID: 1299509)	new	33

2 Functional deviations

2.1 [067]ConfigureFmInterrupt API assumes a parameter with 8 bytes boundary, but actual boundary is 4 bytes

Description

STATUS_ADDR_PROTECTED will be returned if ConfigureFmInterrupt API is called with arguments stored in SRAM with 4 bytes boundary of available SRAM or protected boundary SRAM.

Parameters affected

N/A

Trigger condition(s)

Call ConfigureFmInterrupt API with arguments stored in SRAM at 4 bytes boundary of available SRAM or protected boundary of SRAM.

Scope of impact

ConfigureFmInterrupt API will fail by returning STATUS_ADDR_PROTECTED error status when called with argument having 4 bytes boundary of available SRAM or protected boundary of SRAM.

Workaround

Allow 4 bytes margin (i.e. assume that API parameter size as 8 and store the arguments) for API (ConfigureFmInterrupt) parameter.

Fix status

No silicon fix planned. Use workaround.

2.2 [068]SMPU/MPU/PPU protection region size is limited to 2 GB

Description

If SMPU/MPU/PPU protection block size is configured for 4 GB (PROT_SMPU_SMPU_STRUCT_ATT0.REGION.SIZE = 31), then during protection check in SROM, the value of the internal uint32 variable will overflow (4G = 0x1 0000 0000). Therefore, SROM assumes the protection size equals zero, and no protection will be applied.

Parameters affected

N/A

Trigger condition(s)

Configure SMPU/MPU/PPU to protect with region size equal to 4 GB or the region size with value 31u.

Scope of impact

If SMPU/MPU/PPU is configured to protect region size of 4 GB, then SROM software does not apply any protection as per the request.

Workaround

Use two protection blocks of region size equal to 2 GB if 4 GB region size protection is required.

Fix status

No silicon fix planned. Use workaround.

2.3 [069]DirectExecute API may return error if called with arguments placed in SRAM memory

Description

If DirectExecute API is called in master PC (other than PC0 or PC1) with arguments in SRAM_SCRATCH_ADDR, then the API will return STATUS_ADDR_PROTECTED status.

Parameters affected

N/A

Trigger condition(s)

Call DirectExecute API with arguments in SRAM_SCRATCH_ADDR and master PC configured > 1..

Scope of impact

DirectExecute API, if called with master PC configured > 1 and arguments in SRAM_SCRATCH_ADDR, the API will return STATUS_ADDR_PROTECTED.

Workaround

Call DirectExecute API with master PC0 or PC1, if arguments are stored in SRAM memory..

Fix status

No silicon fix planned. Use workaround.

2.4 [096] CAN FD RX FIFO top pointer feature does not function as expected

Description

RX FIFO top pointer function calculates the address for received messages in Message RAM by hardware. This address should be re-start back from the start address after reading all messages of RX FIFO n size (n: 0 or 1). However, the address does not re-start back from the start address when RX FIFO n size is set to 1 (CANFD_CH_RXFnC.FnS = 0x01). This results in CPU/DMA to read messages from the wrong address in Message RAM.

Parameters affected

N/A

Trigger condition(s)

RX FIFO top pointer function is used when RX FIFO n size set to 1 element (CANFD_CH_RXFnC.FnS = 0x01).

Scope of impact

Received message cannot be correctly read by using RX FIFO top pointer function, when RX FIFO n size set to 1 element.

Workaround

Any of the following

1. Set RX FIFO n size to 2 or more when using RX FIFO top pointer function.
2. Do not use RX FIFO top pointer function when RX FIFO n size set to 1 element. Instead of RX FIFO top pointer, read received messages from the Message RAM directly.

Fix status

No silicon fix planned. Use workaround.

2.5 [097] CAN FD debug message handling state machine not get reset to Idle state when CANFD_CH_CCCR.INIT is set

Description

If either CANFD_CH_CCCR.INIT bit is set by the Host or when the M_TTCAN module enters BusOff state, the debug message handling state machine stays in its current state instead of being reset to Idle state. Configuring the bit CANFD_CH_CCCR.CCE does not change CANFD_CH_RXF1S.DMS.

Parameters affected

N/A

Trigger condition(s)

Either CANFD_CH_CCCR.INIT bit is set by the Host or when the M_TTCAN module enters BusOff state.

Scope of impact

The errata is limited to the use case when the Debug on CAN functionality is active. Normal operation of CAN module is not affected, in which case the debug message handling state machine always remains in Idle state. In the described use case, the debug message handling state machine is stopped and remains in the current state signaled by the bit CANFD_CH_RXF1S.DMS. In case CANFD_CH_RXF1S.DMS is set to 0b11, DMA request remains active. Bosch classifies this as non-critical error with low severity, there is no fix for the IP, Bosch recommends the workaround listed also here.

Workaround

In case the debug message handling state machine has stopped while CANFD_CH_RXF1S.DMS is 0b01 or 0b10, it can be reset to Idle state by hardware reset or by reception of debug messages after CANFD_CH_CCCR.INIT is reset to zero.

Fix status

No silicon fix planned. Use workaround.

2.6 [098]TPIU peripheral ID mismatch

Description

TPIU peripheral ID indicates that it is M3-TPIU instead of M4-TPIU.

Parameters affected

N/A

Trigger condition(s)

When debugger reads PID registers for component identification.

Scope of impact

The debuggers read the TPIU as M3-TPIU and no other impact other than this.

Workaround

No specific workaround required. Debuggers can use trace features.

Fix status

No silicon fix planned.

2.7 [137]Limitation of work flash reading

Description

1. Work flash can be read via different CPU cores but only one CPU core is assigned for non-correctable ECC error handling
2. Reading 32 bits of work flash on AXI bus can result in ECC error (Only for CM7 core devices)

Parameter affected

N/A

Trigger condition(s)

1. Reading work flash via CPU core and ECC fault interrupt routed to two or more CPU cores
2. Reading 32 bits of work flash via CM7_0/1 or other AXI master and adjacent 32 bits of work flash is not initialized (Only for CM7 core devices)

Scope of impact

1. Only one CPU core can be assigned for non-correctable ECC error handling
2. Work flash should be accessed via AHB or AXI with 64-bit boundary (Only for CM7 core devices)

Workaround

Any of the following:

- Option A (Recommended solution)
  - Problem 1 and problem 2: Set each CPU core to use a separate AHB DMA (M-DMA or P-DMA) channel to read work flash. If non-correctable ECC error occurs, the DMA transaction get aborted and respective CPU core gets the interrupt to manage the non-correctable ECC error
- Option B
  - Problem 1[1)]: Set non-correctable ECC error handling to reset. This way no one CPU core needs to manage the non-correctable ECC error handling
  - Problem 2: Limit work flash data size to 64 bits. Program work flash in units of aligned 64-bit double words and read it as 64-bit double words thru CM7_0/1 or another AXI master (Only for CM7 core devices)
- Option C
  - Problem 1[2)]: Assign one CPU core for non-correctable ECC error handling and that core informs the error to the other core which caused the error, but it takes time
  - Problem 2: Use Option B

Fix status

No silicon fix planned. Use workaround. Infineon FLS and FEE driver were updated with workaround A. TRM was updated for this limitation.

2.8 [138]ROM boot code clears to zero last 2 KB of SRAM

Description

For TRAVEO™ T2G Body Entry devices, ROM boot code clears to zero last 2 KB of SRAM. This region is available to the user after boot. However, data retention across resets is not guaranteed in this area.

For TRAVEO™ T2G Body High devices, ROM boot code clears to zero first word of last 2 KB of SRAM. This region is available to the user after boot. However, data retention across resets is not guaranteed in this area.

Paramater affected

N/A

Trigger condition(s)

After ROM boot

Scope of impact

For TRAVEO™ T2G Body Entry devices, data retention across resets is not guaranteed in last 2 KB of SRAM.

For TRAVEO™ T2G Body High devices, data retention across resets is not guaranteed in first word of last 2 KB of SRAM.

Workaround

For TRAVEO™ T2G Body Entry devices, do not use last 2 KB of SRAM for data retention.

For TRAVEO™ T2G Body High devices, do not use first word of last 2 KB of SRAM for data retention.

Scope

No silicon fix planned. Use workaround. TRM was updated for this limitation.

2.9 [139]Limitation of programming SFlash Normal Access Restrictions (row 13)

Description

CM0+ cache is not disabled when programming SFlash Normal Access Restrictions (row 13) by WriteRow SROM API. Occasionally, writing to SFlash Normal Access Restrictions (row 13) may return error status “0xF00000A4” (ProgramRow is invoked on unerased cells or blank check fails).

Parameter affected

N/A

Trigger condition(s)

WriteRow SROM API is called on Normal Access Restrictions (row 13)

[1] Not recommended to use for EEPROM emulation. EEPROM emulation needs to cope with aborted write/erase. In such a scenario, option B leads to deadlock in endless resets. However, option B can be used if work flash update is not intended in the field.

[2] Not recommended to use with MCAL, since the inter-core communication is too slow.

Scope of impact

Error status – 0xF00000A4 “ProgramRow is invoked on unerased cells or blank check fails” is returned

Workaround

Disable CM0+ cache before call to WriteRow (to SFlash row 13) and enabling the cache back after the SROM API execution. Following sequence could be a recommended sequence:

1. FLASHC_CM0_CA_CTL0.CA_EN = 0; // Disable the CM0+ cache
2. Trigger WriteRow SROM API on NAR (row 13)
3. WriteRow successful
4. FLASHC_CM0_CA_CTL0.CA_EN = 1; // Enable the CM0+ cache

Fix status

No silicon fix planned. Use workaround. TRM was updated for this limitation.

2.10 [147]CAN FD controller message order inversion when transmitting from dedicated Tx Buffers configured with same Message ID

Description

Configuration:

Several Tx Buffers are configured with same Message ID. Transmission of these Tx Buffers is requested sequentially with a delay between the individual Tx requests.

Expected behavior:

When multiple Tx Buffers that are configured with the same Message ID have pending Tx requests, they shall be transmitted in ascending order of their Tx Buffer numbers. The Tx Buffer with lowest buffer number and pending Tx request is transmitted first.

Observed behavior:

It may happen, depending on the delay between the individual Tx requests, that in the case where multiple Tx Buffers are configured with the same Message ID the Tx Buffers are not transmitted in order of the Tx Buffer number (lowest number first).

Parameters affected

N/A

Trigger condition(s)

When multiple Tx Buffers that are configured with the same Message ID have pending Tx requests.

Scope of impact

In the case described it may happen, that Tx Buffers configured with the same Message ID and pending Tx request are not transmitted with lowest Tx Buffer number first (message order inversion).

Workaround

Any of the following:.

1. First write the group of Tx message with the same Message ID to the Message RAM and then afterwards request transmission of all these messages concurrently by a single write access to CANFDx_CHy_TXBAR. Before requesting a group of Tx messages with this Message ID ensure that no message with this Message ID has a pending Tx request.
2. Use the Tx FIFO instead of dedicated Tx Buffers for the transmission of several messages with the same Message ID in a specific order.

Applications not able to use workaround #1 or #2 can implement a counter within the data section of their messages sent with same ID in order to allow the recipients to determine the correct sending sequence.

Fix status

No silicon fix planned. Use workaround.

2.11 [162]Temperature sensor accuracy issue

Description

TRAVEO™T2G Body Entry devices have inconsistent references applied to temperature sensor calibration.

Parameter affected

SID200, SID201, SID201A

Trigger condition(s)

Using the temperature sensor

Scope of impact

It results in inaccurate calculation of measured temperature.

Workaround

Set bit 9, 8 and 6 of PASS_TEST_CTL register (address: 0x409F0080) to 1 while keeping the other bits unchanged, after a reset or DeepSleep wakeup.

Fix status

No silicon fix planned. Use workaround. The architecture TRM (002-19314 Rev. *H) was updated to add the PASS_TEST_CTL register description.

2.12 [167]CAN FD incomplete description of Dedicated Tx Buffers and Tx Queue related to transmission from multiple buffers configured with the same Message ID

Description

The following is the updated description in Section 3.5.2 Dedicated Tx Buffers and 3.5.4 Tx Queue of the Architecture TRM related to transmission from multiple buffers configured with the same Message ID.

3.5.2 Dedicated Tx Buffers

- Wording TRM: If multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.
- Enhancement: These Tx buffers shall be requested in ascending order with lowest buffer number first. Alternatively all Tx buffers configured with the same Message ID can be requested simultaneously by a single write access to CANFDx_CHy_TXBAR.

3.5.4 Tx Queue

- Wording TRM: If multiple queue buffers are configured with the same Message ID, the queue buffer with the lowest buffer number is transmitted first.
- Replacement: In case that multiple Tx Queue buffers are configured with the same Message ID, the transmission order depends on numbers of the buffers where the messages were stored for transmission. As these buffer numbers depend on the then current states of the PUT Index, a prediction of the transmission order is not possible.

- Wording TRM: An Add Request cyclically increments the Put Index to the next free Tx Buffer.
- Replacement: The PUT Index always points to that free buffer of the Tx Queue with the lowest number.

Parameters affected

N/A

Trigger condition(s)

Using multiple dedicated Tx Buffers or Tx Queue Buffers configured with the same Message ID

Scope of impact

In the case the dedicated Tx Buffers with the same Message ID are not requested in ascending order or at the same time or in case of multiple Tx Queue Buffers with the same Message ID, it cannot be guaranteed, that these messages are transmitted in ascending order with lowest buffer number first.

Workaround

In case a defined order of transmission is required the Tx FIFO shall be used for transmission of messages with the same Message ID. Alternatively dedicated Tx Buffers with the same Message ID shall be requested in ascending order with lowest buffer number first or by a single write access to CANFDx_CHy_TXBAR. Alternatively a single Tx Buffer can be used to transmit those messages one after the other.

Fix status

No silicon fix planned. Use workaround.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *H for body controller entry family
- 002-24401 Rev. *F for body controller high family
- 002-33175 Rev. *A for cluster entry family
- 002-25800 Rev. *D for cluster 2D family

2.13 [175]Misleading status is returned for Flash and eFuse system calls if there are pending NC ECC faults in SRAM controller #0

Description

Flash and eFuse system calls will return misleading status of 0xf0000005 (“Page is write protected”) even for non-protected row or 0xf0000002 (“Invalid eFuse address”) for valid eFuse address in case of pending NC ECC faults in SRAM controller #0.

Parameters affected

Return status of Flash and eFuse system calls

Trigger condition(s)

NC ECC fault(s) pending in SRAM controller #0 and SWPUs are populated in the design.

Scope of impact

Flash and eFuse system calls will not work until the NC ECC fault(s) pending in SRAM controller #0 is properly handled.

Workaround

If the NC ECC fault(s) are not due to hardware malfunction (i.e. if the faults are due to usage of non-initialized SRAM or improper SRAM initialization), then clearing of these pending faults will resolve the issue.

Fix status

No silicon fix planned.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *I for body controller entry family
- 002-24401 Rev. *G for body controller high family
- 002-33175 Rev. *A for cluster entry family
- 002-25800 Rev. *D for cluster 2D family

2.14 [176]WDT reset causes loss of SRAM retention

Description

Architecture TRM Table 19-1 shows WDT reset can retain SRAM if there is an orderly shutdown of the SRAM only during a warning interrupt. However, this is wrong. WDT reset causes loss of SRAM retention.

Parameters affected

N/A

Trigger condition(s)

WDT reset

Scope of impact

WDT reset causes loss of SRAM retention.

Workaround

None

Fix status

No silicon fix planned.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *I for body controller entry family
- 002-24401 Rev. *G for body controller high family
- 002-33175 Rev. *A for cluster entry family
- 002-25800 Rev. *D for cluster 2D family

2.15 [185]Crypto ECC errors may be set after boot with application authentication

Description

Due to the improper initialization of the Crypto memory buffer, Crypto ECC errors may be set after boot with application authentication. In spite of the Crypto ECC errors, the result of the authentication is reliable.

Parameters affected

N/A

Trigger condition(s)

Boot device with application authentication

Scope of impact

Crypto ECC errors may be set after boot with application authentication.

Workaround

Clear or ignore Crypto ECC errors which generated during boot with application authentication.

Fix status

No silicon fix planned.
The architecture technical reference manuals were updated:

- 002-19314 Rev. *I for body controller entry family
- 002-24401 Rev. *G for body controller high family
- 002-33175 Rev. *A for cluster entry family
- 002-25800 Rev. *E for cluster 2D family

2.16 [198]Incomplete erase of Code Flash cells could happen Erase Suspend / Erase Resume is used along with Erase Sector operation in Non-Blocking mode

Description

Code Flash memory can be erased in “Non-Blocking” mode; a Non-Blocking mode supported option allow users to suspend an ongoing erase sector operation. When an ongoing erase operation is interrupted using “Erase Suspend” and “Erase Resume”, Flash cells may not have been erased completely, even after the erase operation complete is indicated by FLASHC_STATUS register. Only Code Flash is impacted by this issue, Work Flash and Supervisory Flash (SFlash) are not impacted.

Parameters affected

N/A

Trigger condition(s)

Using EraseSector System Call in Non-Blocking mode for CM0+ to erase Code Flash and the ongoing erase operation is interrupted using EraseSuspend and EraseResume System calls.

Scope of impact

When Code Flash sectors are erased in Non-Blocking mode and the ongoing erase operation is interrupted by Erase Suspend / Erase Resume, it cannot be guaranteed that the Code Flash cells are fully erased. Any read on the Code Flash area after the erase is complete or read on the programmed data after ProgramRow is complete can trigger ECC errors.

Workaround

Any of the following

1. User can use Non-Blocking mode for EraseSector, but users must not interrupt the erase operation using Erase Suspend / Erase Resume.
2. If a Code Flash sector erase operation is interrupted using Erase Suspend / Erase Resume, then erase the same sector again without Erase Suspend / Erase Resume before reading the sector or programming the sector.

Fix status

Fixed to update the Flash settings from the following date code

- CYT2B6/7/9: 304xxxxx
- CYT2BL: 312xxxxx
- CYT3BB/4BB: 240xxxxx
- CYT4BF: 312xxxxx
- CYT2CL: 312xxxxx
- CYT3DL: 312xxxxx
- CYT4DN: 312xxxxx

2.17 [199]Limitation for keeping the port state from peripheral IP after wakeup from DeepSleep

Description

The port state is not retained when the port selects peripheral IP (except for LIN or CAN FD) and MCU wakes up from DeepSleep.

Parameters affected

N/A

Trigger condition(s)

The port selects peripheral IP (except for LIN or CAN FD) and MCU wakes up from DeepSleep.

Scope of impact

Unexpected port output change might affect user system.

Workaround

If the port selects peripherals IP (except for LIN or CAN FD) and the port output value need to keep after wakeup from DeepSleep, set HSIOM_PRTx_PORT_SEL.IOy_SEL = 0 (GPIO) before DeepSleep and set the required output value in GPIO configuration registers. After wakeup, change HSIOM_PRTx_PORT_SEL.IOy_SEL back to the peripheral IP.

Fix status

No silicon fix planned.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *I for body controller entry family
- 002-24401 Rev. *G for body controller high family

- 002-33175 Rev. *B for cluster entry family
- 002-25800 Rev. *E for cluster 2D family

2.18 [201]A part of the PWR_CTL2.BGREF_LPMODE description is lacked in the existing register TRM

Description

The following is lacked from the PWR_CTL2.BGREF_LPMODE description in the existing register TRM.

“This register will not set unless CLK_ILO0_CONFIG.ILO0_ENABLE==1. When changing back to continuous operation, keep ILO0 enabled for at least 5 ILO0 cycles after clearing this bit to allow for internal synchronization.”

Parameters affected

N/A

Trigger condition(s)

Using the PWR_CTL2.BGREF_LPMODE

Scope of impact

PWR_CTL2.BGREF_LPMODE may not be set or cleared.

Workaround

Use the PWR_CTL2.BGREF_LPMODE according to the following description.

“This register will not set unless CLK_ILO0_CONFIG.ILO0_ENABLE==1. When changing back to continuous operation, keep ILO0 enabled for at least 5 ILO0 cycles after clearing this bit to allow for internal synchronization.”

Fix status

No silicon fix planned. Register TRM was updated.

2.19 [202]Limitation of clock configuration before entering DeepSleep mode

Description

DeepSleep should not be entered while any FLL/PLL is enabled and using ECO as its reference clock. Since the unstable ECO clock after wakeup is outside the allowed reference clock limits for FLL/PLL, there is possibility of failing the DeepSleep wakeup.

Parameters affected

N/A

Trigger condition(s)

DeepSleep transition while any FLL/PLL is enabled and using ECO as its reference clock.

Scope of impact

There is possibility of failing the DeepSleep wakeup.

Workaround

If any FLL/PLL is operating with the ECO as its reference clock, change the clock to either ECO direct or IMO direct or IMO with FLL/PLL before entering DeepSleep.

Fix status

No silicon fix planned.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *I for body controller entry family
- 002-24401 Rev. *G for body controller high family
- 002-33175 Rev. *B for cluster entry family
- 002-25800 Rev. *E for cluster 2D family

2.20 [203]Several data retention information in Register TRM are incorrect

Description

The following registers are described as ‘Retained’ in the Register TRM while it is not guaranteed that the value before entering DeepSleep mode is still readable from the register.

- SARADC: PASSx_SARy_CHz_RESULT
- SRSS: PWR_LVD_STATUS
- SRSS: PWR_LVD_STATUS2
- SRSS: CLK_CAL_CNT1
- SRSS: CLK_CAL_CNT2
- SRSS: CLK_FLL_STATUS
- SRSS: WDT_INTR
- SRSS: WDT_INTR_MASKED
- SRSS: CLK_PLL400Mx_STATUS (not for CYT2B6/7/9/L)
- MIXER: MIXER_DST_STRUCT_INTR_DST_MASKED (only for cluster devices)

Parameters affected

N/A

Trigger condition(s)

Use of the related function and wakeup from DeepSleep mode.

Scope of impact

The values before entering DeepSleep are not retained.

Workaround

For PASSx_SARy_CHz_RESULT, any of the following:

1. Store the conversion values at another memory location before entering DeepSleep mode
2. Restart the conversion after wakeup from DeepSleep mode

For the other registers:

- Rewrite the register value or read the status flags again after wakeup

Fix status

No silicon fix planned. Register TRM was updated.

2.21 [204]SCBx_INTR_TX.UNDERFLOW bit may be set unintentionally

Description

There is possibility of setting the SCBx_INTR_TX.UNDERFLOW bit even if the FIFO is not empty.

Parameters affected

N/A

Trigger condition(s)

Using the TX FIFO for SCB when the AHB-Lite interface clock (CLK_GR6) frequency of the AHB bus is greater than 3x the SCB functionality clock (PCLK_SCBx_CLOCK).

Scope of impact

SCBx_INTR_TX.UNDERFLOW bit may be set unintentionally.

Workaround

Ignore the SCBx_INTR_TX.UNDERFLOW bit if the FIFO is not empty.

Fix status

No silicon fix planned. Register TRM was updated.

2.22 [206] Hardfault may occur when the SROM APIs listed below while executing EraseSector or ProgramRow in non-blocking mode

Description

The following SROM APIs read data from bank#0 (or bank#1 if dual bank mode with mapping B is used) in SFlash. While doing that the check for active non-blocking erase or program of bank#0 (or bank#1 if dual bank mode with mapping B is used) is not performed. Therefore, reading bank#0 (or bank#1 if dual bank mode with mapping B is used) while there is an active erase/program operation triggers a bus error. This results in a hardfault occurrence based on FLASHC_FLASH_CTL register settings.

Affected SROM APIs:

- ReadSWPU
- WriteSWPU
- GenerateHash
- Checksum[3)]
- ComputeBasicHash[3)]
- CheckFactoryHash
- ProgramWorkFlash[4)]
- SwitchOverRegulators (not for CYT2B6/7/9/L, CYT2CL)
- LoadRegulatorsTrims (not for CYT2B6/7/9/L, CYT2CL)

3 Do not call it to calculate on the bank where programming/erasing is in progress
4 Do not use it during non-blocking operation

Parameters affected

N/A

Trigger condition(s)

Calling the affected SROM APIs while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).

Scope of Impact

The affected SROM APIs cannot be used while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).

Workaround

Do not use the affected SROM APIs while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).

Fix status

No silicon fix planned.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *J for body controller entry family
- 002-24401 Rev. *H for body controller high family
- 002-33175 Rev. *D for cluster entry family
- 002-25800 Rev. *F for cluster 2D family

Impact on Infineon software

Impact: Limitation

Related modules: S-LLD, HSM-Perf-Lib

Comment: While executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used), users must not do anything of following:

1. Call CySldProt_GetSwpuFlashStructCfg
2. Call CySldProt_VerifySecureDomainFlashWriteProtection if CySldProt_SwpuFlashStructGroupConfigurations is non-empty

2.23 [209] CAN FD sporadic data corruption (payload) in case acceptance filtering is not finished before reception of data R3 (DB7..DB4) is completed

Description

During frame reception the Rx Handler accesses the external Message RAM for acceptance filtering (read accesses) and for storing of the accepted messages (write accesses).

The time needed for acceptance filtering and for storing of a received message depends on

- the Host clock frequency
- the worst-case latency of the read and write accesses to the external Message RAM
- the number of configured filter elements
- the workload of the transmit message (Tx) handler in parallel to the receive message (Rx) handler

Received data bytes (DB0..DBm) from the CAN Core are buffered in the cache of the Rx Handler before they are written to the Message RAM (in words of 4 byte). Data words inside the Message RAM are numbered from R2 to Rn (n ≤ 17).

2 Functional deviations

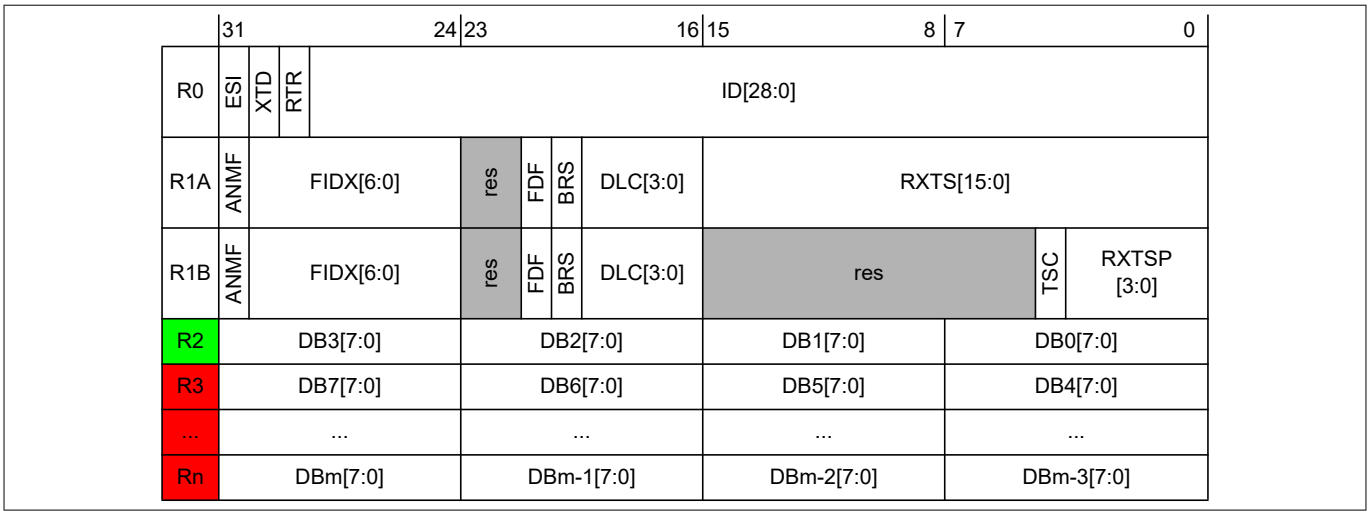


Figure 1 Rx Buffer and FIFO Element

Under the following conditions a received message will have corrupted data while the received message is signaled as valid to the host.

1. The data length code (DLC) of the received Message is greater than 4 (DLC > 4)
2. The storage of Ri of a received message into the Message RAM (after acceptance filtering is done) has not completed before R(i+1) is transferred from the CAN Core into the cache of the Rx Handler (where 2 ≤ i ≤ 5)
3. While condition 1 and 2 apply, a concurrent read of data word Ri from the cache and write of data word R(i+1) into the cache of the Rx handler happens

The data will be corrupted in a way, that in the Message RAM R(i+1) has the same content as Ri.

Despite the corrupted data, the M_TTCAN signals the storage of a valid frame in the Message RAM:

- Rx FIFO: FIFO put index RXFnS.FnPI is updated
- Dedicated Rx Buffer: New Data flag NDATn.NDxx is set
- Interrupt flag IR.MRAF is not set

The issue may occur in FD Frame Format as well as in Classic Frame Format.

Figure 2 shows how the available time for acceptance filtering and storage is reduced.

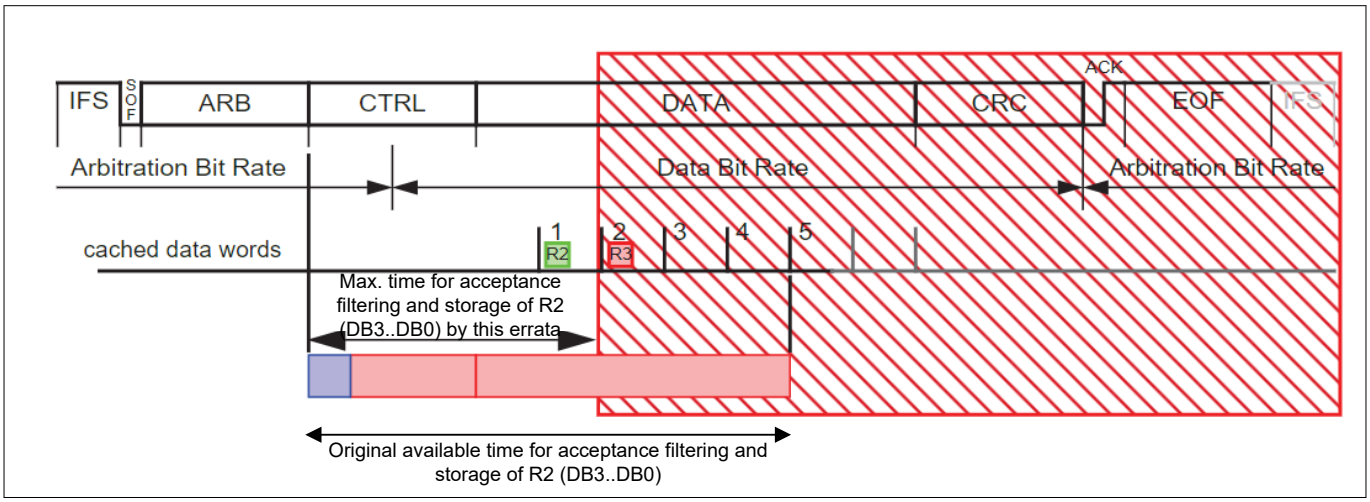


Figure 2 CAN frame with DLC > 4

Table 2 TRAVEO™ T2G: Minimum host clock frequency for CAN FD when DLC = 5

Number of configured active filter element 11-bit IDs/29-bit IDs[1) 2)]	Number of active CAN channels in an instance [3)]	Arbitration bit rate = 0.5 Mbps				Arbitration bit rate = 1 Mbps			
		Data bit rate = 0.5 Mbps	Data bit rate = 1 Mbps	Data bit rate = 2 Mbps	Data bit rate = 4 Mbps	Data bit rate = 1 Mbps	Data bit rate = 2 Mbps	Data bit rate = 4 Mbps	Data bit rate = 5 Mbps
32/16	2	3.9 MHz	7.1 MHz	13.1 MHz	22.8 MHz	7.7 MHz	14.1 MHz	26.1 MHz	31.5 MHz
	3	5.4 MHz	9.9 MHz	18.3 MHz	31.8 MHz	10.7 MHz	19.7 MHz	36.5 MHz	44.0 MHz
	4	6.9 MHz	12.7 MHz	23.5 MHz	40.8 MHz	13.8 MHz	25.3 MHz	46.9 MHz	56.5 MHz
	5	8.4 MHz	15.5 MHz	28.6 MHz	49.9 MHz	16.8 MHz	30.9 MHz	57.2 MHz	69.0 MHz
64/32	2	7.4 MHz	13.5 MHz	24.9 MHz	43.4 MHz	14.7 MHz	26.9 MHz	49.8 MHz	60.0 MHz
	3	10.3 MHz	18.8 MHz	34.9 MHz	60.7 MHz	20.5 MHz	37.6 MHz	69.7 MHz	84.0 MHz[4)]
	4	13.2 MHz	24.2 MHz	44.8 MHz	78.0 MHz	26.3 MHz	48.4 MHz	89.5 MHz[4)]	107.9 MHz[5)]
	5	16.1 MHz	29.6 MHz	54.7 MHz	95.3 MHz[4)]	32.1 MHz	59.1 MHz	109.4 MHz[5)]	131.8 MHz[5)]
96/48	2	10. 8 MHz	19.9 MHz	36.8 MHz	64.0 MHz	21.6 MHz	39.7 MHz	73.5 MHz	88.6 MHz[4)]
	3	15.1 MHz	27.8 MHz	51.5 MHz	89.6 MHz[4)]	30.2 MHz	55.6 MHz	102.9 MHz[5)]	124.0 MHz[5)]
	4	19.4 MHz	35.7 MHz	66.1 MHz	115.1 MHz[5)]	38.8 MHz	71.4 MHz	132.2 MHz[5)]	159.3 MHz[5)]
	5	23.7 MHz	43.6 MHz	80.8 MHz[4)]	140.7 MHz[5)]	47.4 MHz	87.2 MHz[4)]	161.5 MHz[5)]	194.7 MHz[5)]
128/64	2	14.3 MHz	26.3 MHz	48.6 MHz	84.7 MHz[4)]	28.4 MHz	52.5 MHz	97.2 MHz[4)]	117.2 MHz[5)]
	3	20.0 MHz	36.8 MHz	68.0 MHz	118.5 MHz[5)]	40.0 MHz	73.5 MHz	136.0 MHz[5)]	164.0 MHz[5)]
	4	25.7 MHz	47.2 MHz	87.5 MHz[4)]	152.3 MHz[5)]	51.4 MHz	94.4 MHz[4)]	174.9 MHz[5)]	210.8 MHz[5)]
	5	31.4 MHz	57.7 MHz	106.9 MHz[5)]	186.1 MHz[5)]	62.7 MHz	115.4 MHz[5)]	213.7 MHz[5)]	257.5 MHz[5)]

1) M_TTCAN starts always at filter element #0 and proceeds through the filter list to find a matching element. Acceptance filtering stops at the first matching element and the following filter elements are not evaluated for this message. Therefore, the sequence of configured filter elements has a significant impact on the performance of the filtering process.

2) Acceptance filtering search for 11-bit IDs and 29-bit IDs filter element is running separately, only one configured filter setting should be considered. Searching for one 29-bit filter element requires approximately double cycles for one 11-bit filter element.

3) See the device datasheet for the supported number of channels

4) Frequency is not reachable since the maximum host clock frequency for M_TTCAN in CYT2B6 is 80 MHz

5) Frequency is not reachable since the maximum host clock frequency for M_TTCAN in all TRAVEO™ T2G is 100 MHz

2 Functional deviations

Parameters Affected

N/A

Trigger condition(s)

Under the following conditions a received message will have corrupted data while the received message is signaled as valid to the host.

1. The data length code (DLC) of the received Message is greater than 4 (DLC > 4)
2. The storage of Ri of a received message into the Message RAM (after acceptance filtering is done) has not completed before R(i+1) is transferred from the CAN Core into the cache of the Rx Handler (where 2 ≤ i ≤ 5)
3. While condition 1 and 2 apply, a concurrent read of data word Ri from the cache and write of data word R(i+1) into the cache of the Rx handler happens

Scope of impact

The erratum is limited to the case when the Host clock frequency used in the actual device is below the limit shown in Table 2.

Corrupted data is written to the Rx FIFO element respective the dedicated Rx Buffer.

The received frame is nevertheless signaled as valid.

Workaround

Check whether the minimum Host clock frequency, that is shown in Table 2, is below the Host clock frequency used in the actual device.

If yes, there is no problem with the selected configuration.

If no, use one of the following two workarounds.

<First workaround>

Try different configuration by changing the following parameters until the actual host clock frequency (CLK_GR5) is above the minimum host frequency shown in Table 2.

- Increase the CLK_GR5 frequency in the actual device
- Reduce the CAN-FD Data Bit rate
- Reduce the number of configured filter elements
- Reduce the number of active CAN channels in an instance

Also, use DLC>=8 instead of DLCs 5, 6, and 7 in the CAN Environment/System, as they place higher demands on the minimum Host clock frequency (the worst case is DLC=5) or restrict your CAN Environment/System to DLC 4.

Note: While changing the actual host clock frequency, CLK_GR5 must always be equal or higher than PCLK_CANFD[x]_CLOCK_CAN[y] for all configurations.

<Second workaround>

Due to condition 3 the issue occurs only sporadically. Use an end-to-end (E2E) protection (for example, checksum or CRC covering the data field) and add it to all messages in the CAN system, to detect data corruption in received frames.

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: Limitation

Related modules: CAN, MCU

Comment: The user must evaluate the impact of the erratum for each CAN instance separately. A CAN instance is the entirety of CanControllers with the same CanControllerInstance value.

1. For the number of active CAN nodes: Use the maximum number of CanController configurations of a CAN instance that can be active (Autosar controller state STARTED or SLEEP) at a time
2. For the host clock frequency: In McuPeriGroupSettings locate the setting with McuPeriGroup=MCU_PERI_GROUP5_MMIO5 and take the value from McuPeriGroupClockFrequency
3. For the number of configured active filter element 11-bit IDs/29-bit IDs: Use the corresponding values from the “Message RAM (…) linking table” in the generated Can_PBcfg.h file. Note that each CanController has its separate table. Take the maximum values.
4. For the Arbitration bit rate: Use the maximum CanControllerBaudRate value of all the CanControllers
5. For the Data bit rate: Use the maximum CanControllerFdBaudRate value of all the CanControllers if configured. Otherwise use CanControllerBaudRate.

2.24 [212] Description for PASS SARx to TCPWMx direct connect triggers one-to-one is incorrect in datasheet

Description

The body controller device's datasheet showed ‘trig=2’ in the description for PASS SARx to TCPWMx direct connect triggers one-to-one, which was incorrect as TCPWM’s input trigger selection (TR_IN_SEL) value. The correct value is ‘4’ as shown in the architecture TRM chapter 25 descriptions and table 25-2.

The cluster device's datasheet showed ‘trig=0’ in the description for PASS SARx to TCPWMx direct connect triggers one-to-one, which was incorrect as TCPWM’s input trigger selection (TR_IN_SEL) value. The correct value is ‘2’. Therefore, the correct description and table 25-2 in the architecture TRM chapter 25 are as follows.

Table 25-2 shows how the multiplexer should be handled for the input trigger event generation. The TRAVEO™ T2G Cluster MCU supports the following input triggers:

- Number of specific one-to-one trigger inputs: 1
- Number of general-purpose trigger inputs: 60

Table 3 Handling input trigger multiplexers for cluster devices

Input trigger selection	Input trigger	Input trigger source
0	Constant ‘0’	Constant ‘0’
1	Constant ‘1’	Constant ‘1’
2	HSIOM column ACT#2 or PASS (programmable analog subsystem), through 1:1 trigger mux #2	Refer to the "Alternate function pin assignments" or "Triggers one-to-one" section in the device datasheet
3	tr_all_cnt_in[0]	Refer to the trigger mux block in the device datasheet
:	:	:
62	tr_all_cnt_in[59]	Refer to the trigger mux block in the device datasheet

Parameters affected

N/A

Trigger condition(s)

Using the triggers one-to-one for PASS SARx to TCPWMx direct connect

Scope of impact

The triggers one-to-one for PASS SARx to TCPWMx direct connect cannot work if TCPWM's input trigger selection is not correct.

Workaround

For body devices, use '4' as TCPWM's input trigger selection (TR_IN_SEL) value for PASS SARx to TCPWMx direct connect.

For cluster devices, use '2' as TCPWM's input trigger selection (TR_IN_SEL) value for PASS SARx to TCPWMx direct connect.

Fix status

No silicon fix planned. Datasheet was updated.

Impact on Infineon Software

Impact: No

Related modules: PWM

Comment: MCAL PWM module does not support one-to-one triggers.

2.25 [218]Greater change of Low voltage detection (LVD) level can issue an Over voltage detection (OVD) reset

Description

OVD and LVD share the common resistor ladder for reference voltages. Therefore, greater change of LVD level can cause greater voltage fluctuation at OVD reference voltage leading to OVD reset.

Parameter affected

N/A

Trigger condition(s)

Change of LVD trip selection bits (PWR_LVD_CTL/2.HVLVD1/2_TRIPSEL_HT) by a step width of more than 1 at V_{DDD} > 4.5 V.

Scope of impact

Greater change of LVD level can result in an OVD reset.

Workaround

Any of the following

1. Do not change LVD trip selection bits (PWR_LVD_CTL/2.HVLVD1/2_TRIPSEL_HT) from default setting 0.
2. Increment LVD trip selection bits (PWR_LVD_CTL/2.HVLVD1/2_TRIPSEL_HT) by a step width of 1 per 10 µs. Change LVD1 or LVD2 independently, not at the same time.

Fix status

No silicon fix planned. Use workaround.

The architecture technical reference manuals were updated:

- 002-19314 Rev. *J for body controller entry family
- 002-24401 Rev. *H for body controller high family

- 002-33175 Rev. *D for cluster entry family
- 002-25800 Rev. *F for cluster 2D family

Impact on Infineon software

Impact: Limitation

Related modules: MCU

Comment: Workaround (2) was implemented in MCAL MCU driver version 1.24. Users of older versions can apply one of the suggested workarounds in following ways: Leave McuHvLvdThreshold at its default value of MCU_HVLVD_THRESHOLD_2_8V_TO_2_825V, or apply MCU configurations with changes of McuHvLvdThreshold in steps of 0.1V only.

2.26 [229]System calls: improper usage of memory region end address in control logic

Description

Erroneous system calls execution status will be returned when passing parameters to system calls via SRAM region at the end of available SRAM.

Parameters affected

N/A

Trigger condition(s)

When SRAM_SCRATCH_DATA_ADDR for the following system calls end in the last available SRAM word:

- ProgramRow
- ProgramRow2 (only for CYT6BJ)
- ProgramWorkFlash
- ProgramWorkFlash2 (only for CYT6BJ)

Scope of impact

0xF0000013 (invalid arguments location) will be returned when passing parameters to system calls via SRAM region at the end of available SRAM.

Workaround

Do not use the last word of available SRAM for passing SRAM_SCRATCH_DATA_ADDR for the following system calls:

- ProgramRow
- ProgramRow2 (only for CYT6BJ)
- ProgramWorkFlash
- ProgramWorkFlash2 (only for CYT6BJ)

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: Limitation

Related modules: FLS

Comment: Users of MCAL FLS must make sure that Fls_WriteData is not located immediately below the highest SRAM address. You can use memory mapping via FLS_START_SEC_VAR_NO_INIT_UNSPECIFIED and/or linker configuration to modify the location of the affected object.

2.27 [242]RTC busy status bit stuck issue

Description

RTC busy status flag (BACKUP_STATUS.RTC_BUSY register bit) is abnormally stuck at HIGH even after configuring the RTC user registers and BACKUP_RTC_RW.WRITE register bit is cleared by the software.

Parameter affected

N/A

Trigger condition(s)

RTC user register write operation randomly triggers this problem.

Scope of impact

RTC operation may not work as expected. If the software polls BACKUP_STATUS.RTC_BUSY flag before moving to any other tasks, it will be stuck there.

Workaround

Use the following software workaround:

1. If BACKUP_STATUS.RTC_BUSY flag is set, then run the following dummy write sequence
2. Start the sequence for maximum 8 retries (‘retryCount’) and start with 1
   - **a.** Write ‘1’ to BACKUP_RTC_RW.WRITE register for 'retryCount' times
   - **b.** Write ‘0’ to BACKIP_RTC_RW.WRITE register
   - **c.** Wait for 6 CLK_LF cycles (ILO0 or WCO)
   - **d.** If BACKUP_STATUS.RTC_BUSY flag is ‘0’, exit the loop
   - **e.** If ‘retryCount’ reaches to ‘8’, then handle this error according to user system requirement for failure case unrelated to this errata (e.g. write '1' to BACKUP_RESET.RESET register) else run dummy write sequence from “a”. Workaround for this errata needs to maximum 8 retries.

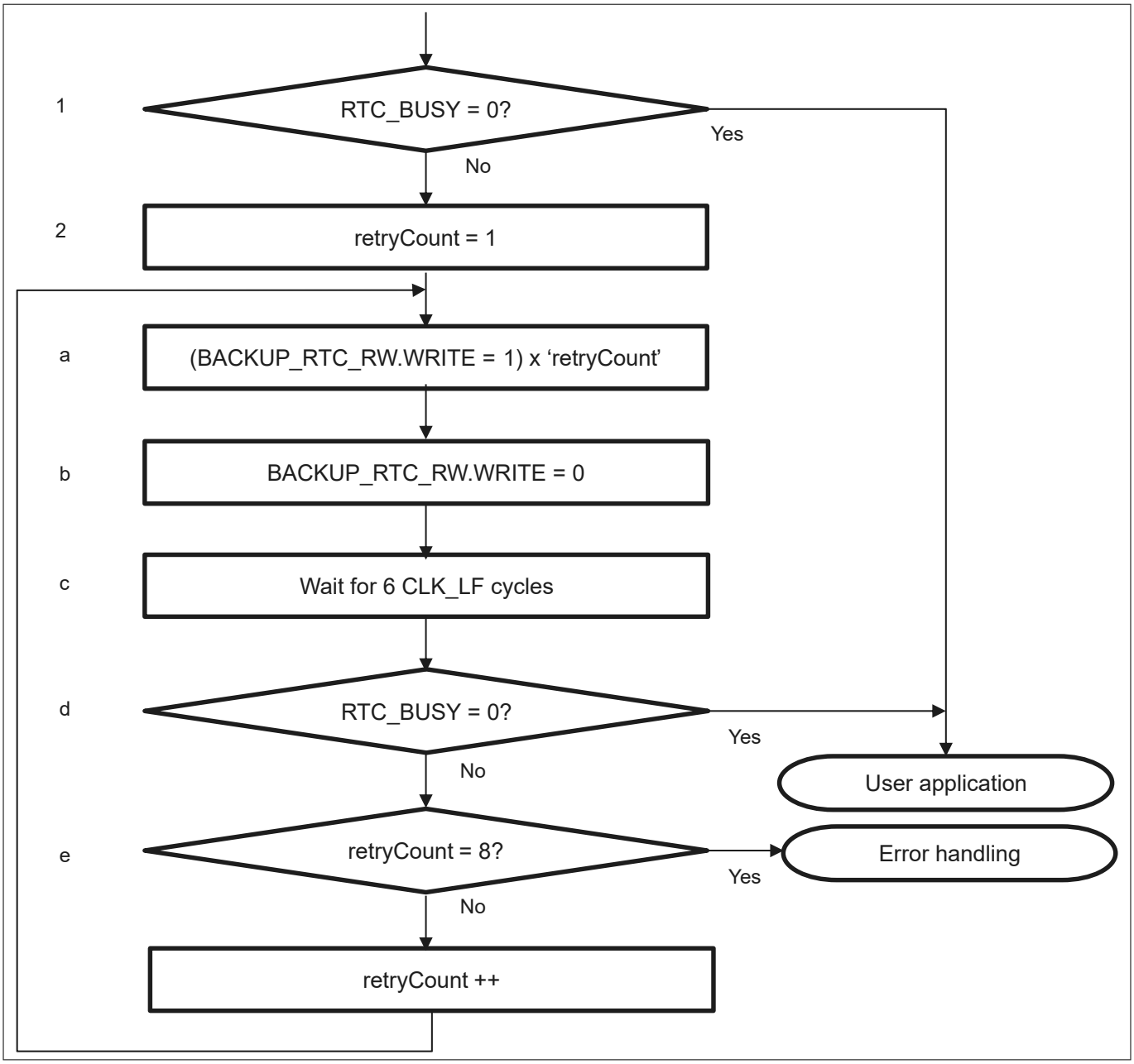


Figure 3 RTC software workaround

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: None
Related modules: N/A
Comments: Software in scope does not access RTC registers.

2.28 [740]Interrupted loads to SP can cause erroneous behavior (Arm® CM4 errata ID: 752770)

Description

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions that can result in the load transaction being repeated are:

1. LDR SP,[Rn],#imm
2. LDR SP,[Rn,#imm]!
3. LDR SP,[Rn,#imm]
4. LDR SP,[Rn]
5. LDR SP,[Rn,Rm]

The affected instructions that can result in the stack-pointer being loaded from an incorrect memory address are:

1. LDR SP,[Rn],#imm
2. LDR SP,[Rn,#imm]!

Parameter affected

N/A

Trigger condition(s)

1. An LDR is executed, with SP/R13 as the destination
2. The address for the LDR is successfully issued to the memory system
3. An interrupt is taken before the data has been returned and written to the stack-pointer

Scope of impact

Unless the load is being performed to Device or Strongly-Ordered memory, there should be no implications from the repetition of the load. In the unlikely event that the load is being performed to Device or Strongly-Ordered memory, the repeated read can result in the final stack-pointer value being different than had only a single load been performed.

Interruption of the two write-back forms of the instruction can result in both the base register value and final stack-pointer value being incorrect. This can result in apparent stack corruption and subsequent unintended modification of memory.

Workaround

Both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

If repeated reads are acceptable, then the base-update issue may be worked around by performing the stack-pointer load without the base increment followed by a subsequent ADD or SUB instruction to perform the appropriate update to the base register.

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: No

Related modules: None

Comment: Software in scope does not perform stack manipulation. Other software, e.g. OS, may be affected if the erratum's conditions are met.

2.29 [741]VDIV or VSQRT instructions might not complete correctly when very short ISRs are used (Arm® CM4 errata ID: 776924)

Description

On Cortex®-M4 with FPU, the VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

Parameter affected

N/A

Trigger condition(s)

1. The floating point unit is present and enabled
2. Lazy context saving is not disabled
3. A VDIV or VSQRT is executed
4. The destination register for the VDIV or VSQRT is one of s0 - s15
5. An interrupt occurs and is taken
6. The interrupt service routine being executed does not contain a floating point instruction
7. 14 cycles after the VDIV or VSQRT is executed, an interrupt return is executed

A minimum of 12 of these 14 cycles are utilized for the context state stacking, which leaves 2 cycles for instructions inside the interrupt service routine, or 2 wait states applied to the entire stacking sequence (which means that it is not a constant wait state for every access).

In general this means that if the memory system inserts wait states for stack transactions then this erratum cannot be observed.

Scope of impact

The VDIV or VQSRT instruction does not complete correctly and the register bank and FPSCR are not updated, meaning that these registers hold incorrect, out of date, data.

Workaround

There are two workarounds:

1. Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34)
2. Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: No

Related modules: None

Comment: GFX-DRV does not run on TRAVEO™ T2G MCUs with CM4 core. Other software in scope does not use floating point. Interrupt service routines on TRAVEO™ T2G, if used as intended, take more than 2 cycles because of system interrupt to CPU interrupt mapping and because of the need to clear pending interrupts.

2.30 [742]ITM can deadlock when global timestamping enabled (Arm® CM4 errata ID: 806422)

Description

The Cortex®-M4 processor contains an optional Instrumentation Trace Macrocell (ITM). This can be used to generate trace data under software control, and is also used with the Data Watchpoint and Trace (DWT) module which generates event driven trace. The processor supports global timestamping. This allows count values from a system-wide counter to be included in the trace stream.

When connected directly to a CoreSight funnel (or other component which holds ATREADY low in the idle state), the ITM will stop presenting trace data to the ATB bus after generating a timestamp packet. In this condition, the ITM_TCR.BUSY register will indicate BUSY.

Once this condition occurs, a reset of the Cortex®-M4 is necessary before new trace data can be generated by the ITM.

Timestamp packets which require a 5 byte GTS1 packet, or a GTS2 packet do not trigger this erratum. This generally only applies to the first timestamp which is generated.

Parameter affected

N/A

Trigger condition(s)

- Cortex®-M4 is configured with DWT present
- The ATB is connected directly to a CoreSight Funnel or similar component
- Global timestamping is enabled (ITM_TCR.GTSFREQ != b00)
- The ITM is enabled (ITM_TCR.ITMENA == 1)
- An event which causes a timestamp to be generated occurs
- A 2nd event which causes a timestamp to be generated occurs with only bits [20:0] of the timestamp changing

Scope of impact

If the system is susceptible to this erratum, global timestamps can not be used reliably.

Workaround

There is no software workaround for this erratum. If the device being used is susceptible to this erratum, you must not enable global timestamping.

Fix status

No silicon fix planned.

Impact on Infineon software

Impact: Limitation

Related modules: N/A

Comment: General impact on tracing with global timestamps. No impact on productive software.

2.31 [743]Store immediate overlapping exception return operation might vector to incorrect interrupt (Arm® CM4 errata ID: 838869)

Description

The Cortex®-M4 includes a write buffer that permits execution to continue while a store is waiting on the bus. Under specific timing conditions, during an exception return while this buffer is still in use by a store instruction, a late change in selection of the next interrupt to be taken might result in there being a mismatch between the interrupt acknowledged by the interrupt controller and the vector fetched by the processor.

Parameter affected

N/A

Trigger condition(s)

1. The handler for interrupt A is being executed
2. Interrupt B, of the same or lower priority than interrupt A, is pending
3. A store with immediate offset instruction is executed to a bufferable location
   a. STR/STRH/STRB <Rt>, [<Rn>,#imm]
   b. STR/STRH/STRB <Rt>, [<Rn>,#imm]!
   c. STR/STRH/STRB <Rt>, [<Rn>],#imm
4. Any number of additional data-processing instructions can be executed
5. A BX instruction is executed that causes an exception return
6. The store data has wait states applied to it such that the data is accepted at least two cycles after the BX is executed
   a. Minimally this is two cycles if the store and the BX instruction have no additional instructions between them
   b. The number of wait states required to observe this erratum needs to be increased by the number of cycles between the store and the interrupt service routine exit instruction
7. Before the bus accepts the buffered store data, another interrupt C is asserted which has the same or lower priority as A, but a greater priority than B

Scope of impact

The processor should execute interrupt handler C, and on completion of handler C should execute the handler for B. If the conditions above are met, then this erratum results in the processor erroneously clearing the pending state of interrupt C, and then executing the handler for B twice. The first time the handler for B is executed it will be at interrupt C's priority level. If interrupt C is pended by a level-based interrupt which is cleared by C's handler then interrupt C will be pended again once the handler for B has completed and the handler for C will be executed.

If interrupt C is level based, then this interrupt will eventually become re-pending and subsequently be handled.

If interrupt C is a single pulse interrupt, then there is a possibility that this interrupt will be lost.

Workaround

For software not using the memory protection unit, this erratum can be worked around by setting DISDEFWBUF in the Auxiliary Control Register.

In all other cases, the erratum can be avoided by ensuring a DSB occurs between the store and the BX instruction. For exception handlers written in C, this can be achieved by inserting the appropriate set of intrinsics or inline assembly just before the end of the interrupt function, for example:

ARMCC:

```
    ...
    __schedule_barrier();
    __asm{DSB};
    __schedule_barrier();
}
```

GCC:

```
    ...
    __asm volatile ("dsb 0xf":::"memory");
}
```

Fix status

No silicon fix planned. Use workaround.

Imapct on Infineon software

Impact: No

Related modules: None

Comment: As interrupt handlers of software in scope check the interrupt reason, the repetition of interrupt B does not cause any malfunction. As TRAVEO™ T2G only uses level-triggered interrupts, the handler for interrupt C is finally executed.

2.32 [744]Fused MAC instructions give incorrect results for rare data combinations (Arm® CM4 errata ID: 839676)

Description

The Cortex®-M4 processor includes optional floating-point logic which supports the fused MAC instructions (VFNMA, VFNMS, VFMA, VFMS).

This erratum causes fused MAC operations on certain combinations of operands to result in either or both of the following:

- A result being generated which is one Unit of Least Precision (ULP) greater than the correct result
- Inexact or underflow flags written to the Floating-point Status Control Register, FPSCR, incorrectly

Parameters affected

N/A

Trigger condition(s)

The conditions for this erratum are all of the following:

1. Cortex®-M4 is configured with floating-point and it is enabled in the Coprocessor Access Control Register, CPACR

2. Flush-to-Zero (FZ) mode is not enabled, that is, FPSCR.FZ is clear
3. A fused MAC instruction (VFNMA, VFNMS, VFMA, or VFMS) is executed with all of the following properties:
   a. The addition part of the operation is Unlike Signed Addition (USA). This implies that the combination of the instruction used and the signs of the operands means that a subtraction is being performed
   b. The result of the instruction, before rounding, is subnormal. This implies that the result is smaller in magnitude than 2^{-126}
   c. The significance of the product and the addend operand are the same or differ by one

Scope of impact

If this erratum occurs, then the processor either produces an incorrect result to a computation or fails to run a floating-point exception routine when it should.

***Note**: It is expected that most algorithms only use normalized numbers because:*
- *Subnormal results have low precision*
- *It is easier to avoid underflow*

This erratum does not affect algorithms which only use normalized numbers.

Workaround

Enable FZ mode in the FPSCR.

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: No

Related modules: None

Comment: GFX-DRV does not run on TRAVEO™ T2G MCUs with CM4 core. Other software in scope does not use floating point.

2.33 [745]Processor reset asserted asynchronously could corrupt FPB comparator registers and remap to wrong address (Arm® CM4 errata ID: 1299509)

Description

Normally, the debugger uses the Flash Patch and Breakpoint (FPB) unit for breakpoints or for patching ROM code during debugging. However, you can also use the FPB functionality as a method of in-the-field ROM code patching. On the Cortex®-M4 processor, the processor reset can be asynchronously asserted and this could potentially cause problems if the processor is in the middle of writing to debug components (which are not reset by the processor reset) such as the FPB unit.

Parameter affected

N/A

Trigger condition(s)

1. The processor is programming the FPB to remap an address in ROM to fetch a replacement opcode or vector from an area in RAM

2. The processor is asynchronously reset during the programming of the FPB
3. The data is incorrectly written to the FPB register due to metastability

Scope of impact

In the unlikely event of this occurring it may cause incorrect functional operation of the processor to occur. If the FPB is programmed with incorrect data it may cause the processor to unintentionally patch instructions.

Workaround

The problem does not arise if the FPB is not used to patch instructions.

If the FPB is used to patch instructions, a software workaround is to ensure that the FPB is globally disabled before programming any comparators.

If code exists which programs the FPB other than at reset, the software workaround should include:

1. Disable the FPB
2. Program the individual comparators required
3. Explicitly disable the individual comparators not required
4. Re-enable the FPB

This sequence prevents any corruptly programmed FPB comparators from being activated.

Fix status

No silicon fix planned. Use workaround.

Impact on Infineon software

Impact: No

Related modules: None

Comment: Software in scope does not use FPB.

Revision history of CYT2BL errata sheet

Document version	Date of release	Description of changes
1.0	2020-08-28	Initial release
1.1	2021-02-18	Added errata ID 147
1.2	2021-06-29	Added errata ID 161, 162, 164
1.3	2021-09-16	Updated "Workaround" of errata ID 147. Added errata ID 167, 168, 169.
1.4	2022-01-21	Added errata ID 175, 176.
1.5	2022-07-04	Updated "Fixed status" of errata ID 169. Added errata ID 185.
1.6	2022-08-25	Updated "Description" of errata ID 185. Added errata ID 198, 199.
1.7	2022-12-02	Added errata ID 201, 202, 203, 204
1.8	2023-02-08	Updated the description to 5 ILO0 cycles in errata ID 201. Updated "Description" of errata ID 203. Fixed the register name of errata ID 204. Added errata ID 206.
1.9	2023-06-15	Updated errata ID 206 to add (or bank#1 if dual bank mode with mapping B is used).
2.0	2023-10-17	Updated errata ID 206 to add the affected SROM APIs. Added errata ID 212.
2.1	2024-08-23	Added errata ID 218.
2.2	2025-02-21	Added errata ID 229. Updated the “Impact on Infineon software” of errata ID 218. Removed errata ID 161, 164, 168, 169 because the datasheet was updated before official release.
2.3	2025-03-28	Migrated to Infineon errata template. Consolidated the description for all affected devices (each errata ID 137, 138, 198, 203, 206, 209, 212).
2.4	2025-09-30	Removed "sometimes" from description of errata ID 229. Added errata ID 242 and 740 to 745.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-09-30

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-krz1724911104943

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.