

Device	XMC1400
Marking/Step	EES-AA, ES-AA
Package	PG-VQFN-40/48/64, PG-LQFP-64

Overview

This “Errata Sheet” describes product deviations with respect to the user documentation listed below.

Table 1 Current User Documentation

Document	Version	Date
XMC1400 Reference Manual AA-step	V1.0	Oct. 2015
XMC1400 Data Sheet AA-step	V0.3	Oct. 2015

Make sure that you always use the latest documentation for this device listed in category “Documents” at <http://www.infineon.com/xmc1000>.

Notes

- 1. The errata described in this sheet apply to all temperature and frequency versions and to all memory size and configuration variants of affected devices, unless explicitly noted otherwise.*
- 2. Devices marked with EES or ES are engineering samples which may not be completely tested in all functional and electrical characteristics, therefore they must be used for evaluation only. The specific test conditions for EES and ES are documented in a separate “Status Sheet”.*

Conventions used in this Document

Each erratum is identified by **Module_Marker.TypeNumber**:

- **Module**: Subsystem, peripheral, or function affected by the erratum.
- **Marker**: Used only by Infineon internal.
- **Type**: type of deviation
 - **(none)**: Functional Deviation
 - **P**: Parametric Deviation
 - **H**: Application Hint
 - **D**: Documentation Update
- **Number**: Ascending sequential number. As this sequence is used over several derivatives, including already solved deviations, gaps inside this enumeration can occur.

1 History List / Change Summary

Table 2 History List

Version	Date	Remark
1.0	2015-10	Initial version

Table 3 Errata fixed in this step

Errata	Short Description	Change
- none -		

Table 4 Functional Deviations

Functional Deviation	Short Description	XMC1401	XMC1402	XMC1403	XMC1404	Chg	Pg
ACMP_CM.001	Operating range of the Analog Comparator Reference Divider function		X		X	New	7
ADC_AI.008	Wait-for-Read condition for register GLOBRES not detected in continuous auto-scan sequence	X	X	X	X	New	7
ADC_AI.016	No Channel Interrupt in Fast Compare Mode with GLOBRES	X	X	X	X	New	8
BCCU_CM.008	Linear walk starts with a delay after an aborted linear walk		X		X	New	8
BCCU_CM.009	Dimming level not immediately changed for first dimming operation		X		X	New	9

Table 4 Functional Deviations

Functional Deviation	Short Description	XMC1401	XMC1402	XMC1403	XMC1404	Chg	Pg
CCU_AI.007	Automatic shadow transfer feature does not work when system PCLK is faster than MCLK	X	X	X	X	New	9
CCU8_AI.005	PWM outputs from Compare Channel 2 are disabled in asymmetric mode		X		X	New	10
CPU_CM.002	Watchpoint PC functions can report false execution	X	X	X	X	New	12
CPU_CM.003	Prefetch faulting instructions can erroneously trigger breakpoints	X	X	X	X	New	13
Firmware_CM.002	Calculate Target Level for Temperature Comparison User Routine returns zero for valid temperature input parameter	X	X	X	X	New	14
Firmware_CM.003	BMI installation of CAN BSL with external oscillator is not supported			X	X	New	14
Firmware_CM.004	SSC BSL is not supported	X	X	X	X	New	14
Firmware_CM.005	Last byte of SRAM is not available for ASC BSL	X	X	X	X	New	15
Firmware_CM.006	Header resend not supported for incorrect ASC BSL header byte	X	X	X	X	New	15

Table 4 Functional Deviations

Functional Deviation	Short Description	XMC1401	XMC1402	XMC1403	XMC1404	Chg	Pg
USIC_AI.008	SSC delay compensation feature cannot be used	X	X	X	X	New	16
USIC_AI.014	No serial transfer possible while running capture mode timer	X	X	X	X	New	16
USIC_AI.017	Clock phase of data shift in SSC slave cannot be changed	X	X	X	X	New	16
USIC_AI.018	Clearing PSR.MSLS bit immediately deasserts the SELOx output signal	X	X	X	X	New	17

Table 5 Application Hints

Hint	Short Description	XMC1401	XMC1402	XMC1403	XMC1404	Chg	Pg
ADC_AI.H007	Ratio of Sample Time t_s to SHS Clock f_{SH}	X	X	X	X	New	18
ADC_AI.H010	Disabling the ORWD	X	X	X	X	New	19
BCCU_CM.H001	Additional dimming clocks after dimming curve switch		X		X	New	19
BCCU_CM.H004	Packer threshold (CHCONFIGy.PKTH) accepted values		X		X	New	19

Table 5 Application Hints

Hint	Short Description	XMC1401	XMC1402	XMC1403	XMC1404	Chg	Pg
BCCU_CM.H005	Enable a dimming engine for global dimming		X		X	New	20
MultiCAN_AI.H005	TxD Pulse upon short disable request			X	X	New	20
MultiCAN_AI.H006	Time stamp influenced by resynchronization			X	X	New	20
MultiCAN_AI.H007	Alert Interrupt Behavior in case of Bus-Off			X	X	New	20
MultiCAN_TC.H003	Message may be discarded before transmission in STT mode			X	X	New	21
MultiCAN_TC.H004	Double remote request			X	X	New	22
SCU_CM.H001	Temperature Sensor Functionality	X	X	X	X	New	22
USIC_AI.H004	I2C slave transmitter recovery from deadlock situation	X	X	X	X	New	23

2 Functional Deviations

The errata in this section describe deviations from the documented functional behavior.

ACMP_CM.001 Operating range of the Analog Comparator Reference Divider function

The Analog Comparator Reference Divider function is not available when V_{DDP} is below 3 V. To use this function, V_{DDP} must be between 3 V to 5.5 V.

Workaround

None

ADC_AI.008 Wait-for-Read condition for register GLOBRES not detected in continuous auto-scan sequence

In the following scenario:

- A continuous auto-scan is performed over several ADC groups and channels by the Background Scan Source, using the global result register (GLOBRES) as result target ($GxCHCTRY.RESTBS=1_B$), and
 - The Wait-for-Read mode for GLOBRES is enabled ($GLOBRCR.WFR=1_B$),
- each conversion of the auto-scan sequence has to wait for its start until the result of the previous conversion has been read out of GLOBRES.

When the last channel of the auto-scan is converted and its result written to GLOBRES, the auto-scan re-starts with the highest channel number of the highest ADC group number. But the start of this channel does not wait until the result of the lowest channel of the previous sequence has been read from register GLOBRES, i.e. the result of the lowest channel may be lost.

Workaround

If either the last or the first channel in the auto-scan sequence does not write its result into GLOBRES, but instead into its group result register (selected via bit

GxCHCTRy.RESTBS=0_B), then the Wait-for-Read feature for GLOBRES works correctly for all other channels of the auto-scan sequence.

For this purpose, the auto-scan sequence may be extended by a “dummy” conversion of group x/ channel y, where the Wait-for-Read mode must not be selected (GxRCRy.WFR=0_B) if the result of this “dummy” conversion is not read.

ADC AI.016 No Channel Interrupt in Fast Compare Mode with GLOBRES

In fast compare mode, the compare value is taken from bitfield RESULT of the global result register GLOBRES and the result of the comparison is stored in the respective bit FCR.

The channel event indicating that the input becomes higher or lower than the compare value, is not generated.

The comparison is executed correctly, the target bit is stored correctly, source events and result events are generated.

Workaround

The result bit FCR can be evaluated if the input is higher or lower than the compare value.

BCCU CM.008 Linear walk starts with a delay after an aborted linear walk

If a linear walk is previously aborted, the subsequent linear walk starts with a delay. The maximum delay is one linear clock.

Workaround

None.

BCCU CM.009 Dimming level not immediately changed for first dimming operation

For the first dimming operation, the dimming level is not immediately incremented or decremented upon a shadow bit (DES) assertion.

Workaround

None.

CCU AI.007 Automatic shadow transfer feature does not work when system PCLK is faster than MCLK

Each CCU8 or CCU4 Timer Slice has a set of registers that have an associated shadow transfer mechanism. This shadow transfer mechanism works like a buffer where the application software can load new data, and when needed, request the load of this data into the register (this mechanism of loading new data into the current used register is called shadow transfer).

The shadow transfer can be applied to several data registers: compare values (that control the PWM duty cycle), timer period value or dead-time, in case of CCU8 (CCU4 does not have a dead-time feature).

To request the shadow transfer operation, the application software has two possibilities:

- do a software request by writing a 1_B into a specific bitfield
- or enable an automatic request feature, that will allow the hardware to request the shadow transfer in an automatic way (saving this way one software operation)

The automatic shadow transfer request be enabled for the following conditions:

- after the software writes a value into the shadow period register - enabled by setting the CC8ySTC.ASPC = 1_B (in CCU8) or CC4ySTC.ASPC = 1_B (in CCU4)
- after the software writes a value into the shadow compare register - enabled by setting the CC8ySTC.ASCC1 = 1_B (in CCU8) or CC8ySTC.ASCC2 = 1_B (in CCU8) or CC4ySTC.ASCC = 1_B (in CCU4)

- after the software writes a value into the passive level register - enabled by setting the CC8ySTC.ASLC = 1_B (in CCU8) or CC4ySTC.ASLC = 1_B
- after the software writes a value into the dither shadow register - enabled by setting the CC8ySTC.ASDC = 1_B (in CCU8) or CC4ySTC.ASDC = 1_B
- after the software writes a value into the floating prescaler shadow register - enabled by setting the CC8ySTC.ASFC = 1_B (in CCU8) or CC4ySTC.ASFC = 1_B

None of these automatic shadow transfer features work when the system PCLK frequency is higher than the MCLK frequency.

The PCLK can, in XMC1400 device(s), be configured to be 2x faster than the MCLK by setting the CLKCR.PCLKSEL = 1_B.

Whenever the PCLKSEL = 1_B, then none of the automatic shadow transfer request functions work.

Note that the rest of the shadow transfer mechanism/features are still usable as specified.

Workaround

None

CCU8 AI.005 PWM outputs from Compare Channel 2 are disabled in asymmetric mode

Each CCU8 Timer Slice offers an asymmetric PWM mode. This mode is enabled whenever the CC8yCHC.ASE bitfield is set to 1_B.

If the asymmetric mode is not selected (ASE = 0_B), then each compare channel works independently, and can generate each, two complementary PWM signals. The PWM signal information for compare channel 1 is stored in the CC8yST1. The PWM signal information for compare channel 2 is stored in CC8yST2. CC8yST1 and CC8yST2 can then be forward to the PWM outputs. This is depicted in [Figure 1 a](#)).

If the asymmetric mode is selected, the timer slice will generate two complementary asymmetric PWM signals, using the compare channel 1 for setting the PWM and the compare channel 2 for clearing the PWM. The

asymmetric PWM signal is stored in CC8yST1 and then can be forward to the outputs. This is depicted in **Figure 1 b)**.

The asymmetric mode has a bug, where the PWM signal of Compare Channel 2 is not stored in CC8yST2. This means that from the Compare Channel 2 the timer cannot generate any PWM signal - depicted by the cross on **Figure 1 b)**.

Note that the asymmetric PWM is still stored in CC8yST1 and can be used normally by the application.

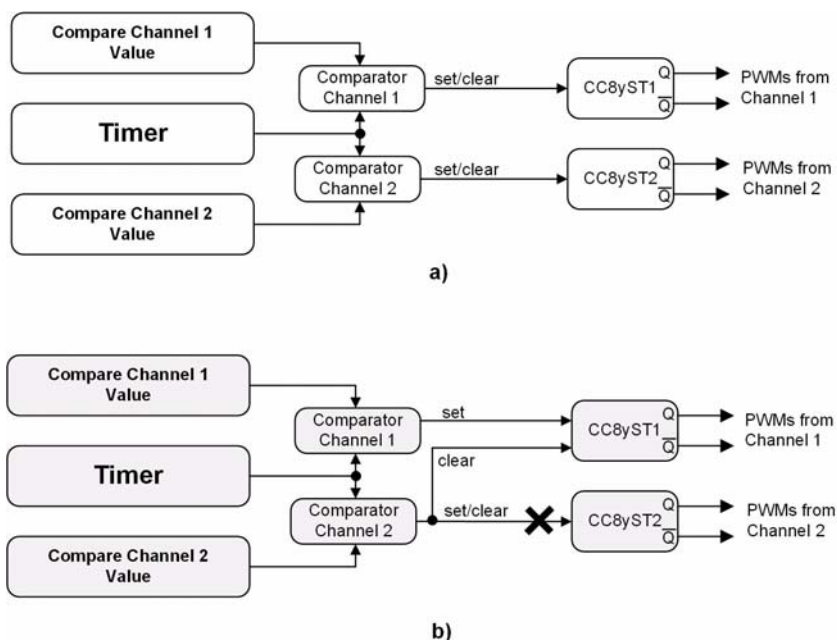


Figure 1 CCU8 simplifier PWM generation scheme - a) normal mode; b) asymmetric mode

Workaround

None

CPU_CM.002 Watchpoint PC functions can report false execution

In the presence of interrupts including those generated by the SVC instruction, it is possible for both the data watchpoint unit's PC match facility and PC sample-register to operate as though the instruction immediately following the interrupted or SVC instruction had been executed.

Conditions

Either:

1. Halting debug is enabled via `C_DEBUGEN = 1`
2. Watchpoints are enabled via `DWTENA = 1`
3. A watchpoint is configured for PC sampling `DWT_FUNCTION = 0x4`
4. The same watchpoint is configured to match a `target instruction`
5. And either:
 - a) The `target instruction` is interrupted before execution, or
 - b) The `target instruction` is preceded by a taken SVC instruction
6. The DWT will unexpectedly match the `target instruction`
7. The processor will unexpectedly enter debug state once inside the exception handler

Or:

1. The debugger performs a read access to the `DWT_PCSR`
2. A `non-committed instruction` is preceded by a taken SVC instruction
3. The `DWT_PCSR` value unexpectedly matches the `non-committed instruction`

Implications

If halting debug is enabled and PC match watchpoints are being used, then spurious entry into halted debug state may occur under the listed conditions.

If the `DWT_PCSR` is being used for coarse grain profiling, then it is possible that the results can include hits for the address of an instruction immediately after an SVC instruction, even if said instruction is never executed.

Workaround

This errata does not impact normal execution of the processor.

A debug agent may choose to handle the infrequent false positive Debug state entry and erroneous PCSR values as spurious events.

CPU_CM.003 Prefetch faulting instructions can erroneously trigger breakpoints

External prefetch aborts on instruction fetches on which a BPU breakpoint has been configured, will cause entry to Debug state. This is prohibited by revision C of the ARMv6-M Architecture Reference Manual. Under this condition, the breakpoint should be ignored, and the processor should instead service the prefetch-abort by entering the HardFault handler.

Conditions

1. Halting debug is enabled via `CDEBUG_EN == '1'`
2. A BPU breakpoint is configured on an instruction in the first 0.5GB of memory
3. The fetch for said instruction aborts via an AHB Error response
4. The processor will erroneously enter Debug state rather than entering HardFault.

Implications

If halting debug is enabled and a BPU breakpoint is placed on an instruction with faults due to an external abort, then a non-compliant entry to Debug state will occur.

Workaround

This errata does not impact normal execution of the processor.

A debug agent may choose to avoid placing BPU breakpoints on addresses that generate AHB Error responses, or may simply handle the Debug state entry as a spurious debug event.

Firmware CM.002 Calculate Target Level for Temperature Comparison User Routine returns zero for valid temperature input parameter

In Calculate Target Level for Temperature Comparison User Routine in Firmware, the temperature sensor threshold value is expected to be returned for a valid range of temperature input parameter of 233K to 388K. This user function typically returns zero value for input parameter out of the valid range, also for some input parameters within the valid range.

Workaround

If user function returns zero for input parameter within the valid range, increase or decrease the input parameter by 1 degree Kelvin in order to use this user function.

Firmware CM.003 BMI installation of CAN BSL with external oscillator is not supported

For the start-up mode selection of CAN BSL and CAN BSL with timeout mode, the Request BMI Installation User Routine with CAN clock source via external oscillator is not supported for EES samples.

Workaround

Use CAN clock source via internal oscillator in the Request BMI Installation User Routine for the CAN BSL start-up modes.

Firmware CM.004 SSC BSL is not supported

SSC Bootstrap Loader start-up mode is not supported for EES and ES samples.

Workaround

None.

Firmware CM.005 Last byte of SRAM is not available for ASC BSL

For start-up mode selection of ASC BSL and ASC BSL with timeout modes, the last byte of available SRAM is not available for user application code. Application length error (BSL_NOK) is transmitted back to the host for application code length of available SRAM size.

Workaround

Host to send application code length of available SRAM size - 1 Byte or less.

Firmware CM.006 Header resend not supported for incorrect ASC BSL header byte

For start-up mode selection of ASC BSL and ASC BSL with timeout modes, the baud rate detection is performed based on the Start Byte (00_H) received from the host. Next, the Header Byte defined in [Table 6](#) is expected from the host. An incorrect Header Byte received leads to a hang-up of the device.

Table 6 Header Byte definition in ASC BSL

Name	Length, Byte	Value	Description
Data sent by the Host:			
BSL_ASC_F	1	6C _H	Header requesting full duplex ASC mode with the current baud rate
BSL_ASC_H	1	12 _H	Header requesting half duplex ASC mode with the current baud rate
BSL_ENC_F	1	93 _H	Header requesting full duplex ASC mode with a request to switch the baud rate
BSL_ENC_H	1	ED _H	Header requesting half duplex ASC mode with a request to switch the baud rate

Workaround

None.

USIC AI.008 SSC delay compensation feature cannot be used

SSC master mode and complete closed loop delay compensation cannot be used. The bit DX1CR.DCEN should always be written with zero to disable the delay compensation.

Workaround

None.

USIC AI.014 No serial transfer possible while running capture mode timer

When the capture mode timer of the baud rate generator is enabled (BRG.TMEN = 1) to perform timing measurements, no serial transmission or reception can take place.

Workaround

None.

USIC AI.017 Clock phase of data shift in SSC slave cannot be changed

Setting PCR.SLPHSEL bit to 1 in SSC slave mode is intended to change the clock phase of the data shift such that reception of data bits is done on the leading SCLKIN clock edge and transmission on the other (trailing) edge.

However, in the current implementation, the feature is not working.

Workaround

None.

USIC AI.018 Clearing PSR.MSLS bit immediately deasserts the SELOx output signal

In SSC master mode, the transmission of a data frame can be stopped explicitly by clearing bit PSR.MSLS, which is achieved by writing a 1 to the related bit position in register PSCR.

This write action immediately clears bit PSR.MSLS and will deassert the slave select output signal SELOx after finishing a currently running word transfer and respecting the slave select trailing delay (T_{td}) and next-frame delay (T_{nf}).

However in the current implementation, the running word transfer will also be immediately stopped and the SELOx deasserted following the slave select delays.

If the write to register PSCR occurs during the duration of the slave select leading delay (T_{ld}) before the start of a new word transmission, no data will be transmitted and the SELOx gets deasserted following T_{td} and T_{nf} .

Workaround

There are two possible workarounds:

- Use alternative end-of-frame control mechanisms, for example, end-of-frame indication with TSCR.EOF bit.
- Check that any running word transfer is completed (PSR.TSIF flag = 1) before clearing bit PSR.MSLS.

3 Application Hints

The errata in this section describe application hints which must be regarded to ensure correct operation under specific application conditions.

ADC_AI.H007 Ratio of Sample Time t_S to SHS Clock f_{SH}

The sample time t_S is programmable to the requirements of the application.

To ensure proper operation of the internal control logic, t_S must be at least four cycles of the prescaled converter clock f_{SH} , i.e. $t_S \geq 4 t_{CONV} \times (DIVS+1)$.

(1) With **SHS*_TIMCFGx.SST > 0**, the sample time is defined by

$$t_S = SST \times t_{ADC}$$

In this case, the following relation must be fulfilled:

- $SST \geq 4 \times t_{CONV}/t_{ADC} \times (DIVS+1)$, i.e. $SST \geq 4 \times f_{ADC}/f_{CONV} \times (DIVS+1)$.

– Example:

with the default setting $DIVS=0$ and $f_{ADC} = f_{MCLK} = 32$ MHz, $f_{SH} = f_{CONV} = 32$ MHz (for $DIVS = 0$):
select $SST \geq 4$.

(2) With **SHS*_TIMCFGx.SST = 0**, the sample time is defined by

$$t_S = (2+STC) \times t_{ADCI}, \text{ with } t_{ADCI} = t_{ADC} \times (DIVA+1)$$

In this case, the following relation must be fulfilled:

- $[(2+STC) \times (DIVA+1)] / (DIVS+1) \geq 4 \times t_{CONV}/t_{ADC} = 4 \times f_{ADC}/f_{CONV}$.

– Example:

With the default settings $STC=0$, $DIVA=1$, $DIVS=0$ and $f_{ADC} = f_{MCLK} = 32$ MHz, $f_{SH} = f_{CONV} = 32$ MHz (for $DIVS = 0$), this relation is fulfilled.

Note: In addition, the condition $f_{ADC} = f_{MCLK} \geq 0.55 f_{SH}$ must be fulfilled.

Note that this requirement is more restrictive than the requirement in ADC_AI.H006.

Definitions

DIVA: Divider Factor for the Analog Internal Clock, resulting from bit field GLOBCFG.DIVA (range: 1..32_D)

DIVS: Divider Factor for the SHS Clock, resulting from bit field SHS*_SHSCFG.DIVS (range: 1..16_D)

STC: Additional clock cycles, resulting from bit field STCS/STCE in registers GxICLASS*, GLOBICLACSSy (range: 0..256_D)

SST: Short Sample Time factor, resulting from bit field SHS*_TIMCFGx.SST (range: 1..63_D)

Recommendation

Select the parameters such that the sample time t_s is at least four cycles of the prescaled converter clock f_{SH} , as described above.

ADC AI.H010 Disabling the ORWD

When the ORWD is disabled while detecting an overvoltage, i.e. while its output signal is high, the output will not always return to zero, but may remain high after disabling.

It is therefore recommended to disable the connected units (ERU, interrupt generation) before disabling the corresponding out-of-range comparator.

BCCU CM.H001 Additional dimming clocks after dimming curve switch

If the dimming curve is switched (from coarse to fine or vice versa), the next dimming process takes additional dimming clocks.

BCCU CM.H004 Packer threshold (CHCONFIGy.PKTH) accepted values

CHCONFIGy.PKTH is defined as 3-bits wide. However, only values 1-4 are accepted.

BCCU CM.H005 Enable a dimming engine for global dimming

When using global dimming as the source of dimming input (CHCONFIG.DSEL = 111_B), enable at least one of the dimming engines (DEEN != 0).

MultiCAN AI.H005 TxD Pulse upon short disable request

If a CAN disable request is set and then canceled in a very short time (one bit time or less) then a dominant transmit pulse may be generated by MultiCAN module, even if the CAN bus is in the idle state.

Example for setup of the CAN disable request:
CAN_CLC.DISR=1 and then CAN_CLC.DISR=0

Workaround

Set all INIT bits to 1 before requesting module disable.

MultiCAN AI.H006 Time stamp influenced by resynchronization

The time stamp measurement feature is not based on an absolute time measurement, but on actual CAN bit times which are subject to the CAN resynchronization during CAN bus operation. The time stamp value merely indicates the number of elapsed actual bit times. Those actual bit times can be shorter or longer than nominal bit time length due to the CAN resynchronization events.

Workaround

None.

MultiCAN AI.H007 Alert Interrupt Behavior in case of Bus-Off

The MultiCAN module shows the following behavior in case of a bus-off status:

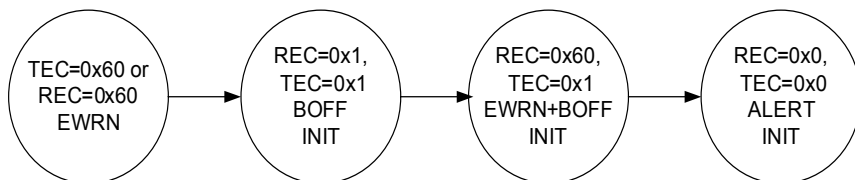


Figure 2 Alert Interrupt Behavior in case of Bus-Off

When the threshold for error warning (EWRN) is reached (default value of Error Warning Level EWRN = 0x60), then the EWRN interrupt is issued. The bus-off (BOFF) status is reached if $TEC > 255$ according to CAN specification, changing the MultiCAN module with REC and TEC to the same value 0x1, setting the INIT bit to 1_B, and issuing the BOFF interrupt. The bus-off recovery phase starts automatically. Every time an idle time is seen, REC is incremented. If REC = 0x60, a combined status EWRN+BOFF is reached. The corresponding interrupt can also be seen as a pre-warning interrupt, that the bus-off recovery phase will be finished soon. When the bus-off recovery phase has finished (128 times idle time have been seen on the bus), EWRN and BOFF are cleared, the ALERT interrupt bit is set and the INIT bit is still set.

MultiCAN TC.H003 Message may be discarded before transmission in STT mode

If $MOFCR_n.STT=1$ (Single Transmit Trial enabled), bit TXRQ is cleared (TXRQ=0) as soon as the message object has been selected for transmission and, in case of error, no retransmission takes places.

Therefore, if the error occurs between the selection for transmission and the real start of frame transmission, the message is actually never sent.

Workaround

In case the transmission shall be guaranteed, it is not suitable to use the STT mode. In this case, $MOFCR_n.STT$ shall be 0.

MultiCAN_TC.H004 Double remote request

Assume the following scenario: A first remote frame (dedicated to a message object) has been received. It performs a transmit setup (TXRQ is set) with clearing NEWDAT. MultiCAN starts to send the receiver message object (data frame), but loses arbitration against a second remote request received by the same message object (NEWDAT will be set).

When the appropriate message object (data frame) triggered by the first remote frame wins the arbitration, it will be sent out and NEWDAT is not reset. This leads to an additional data frame, that will be sent by this message object (clearing NEWDAT).

There will, however, not be more data frames than there are corresponding remote requests.

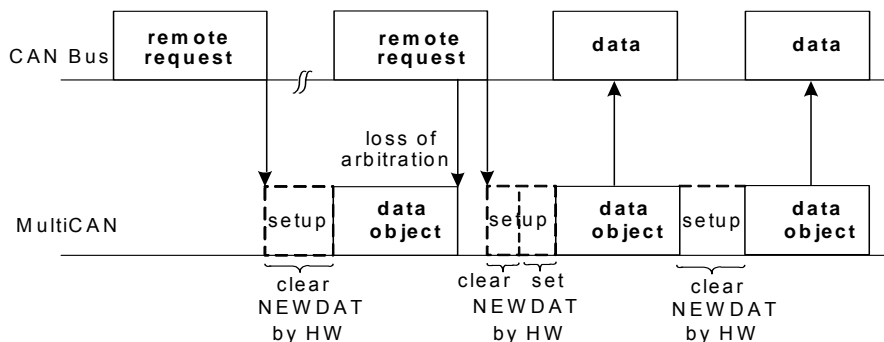


Figure 3 Loss of Arbitration

SCU_CM.H001 Temperature Sensor Functionality

EES samples are not temperature tested, therefore the temperature sensor functionality is not supported.

Workaround

None

USIC AI.H004 I2C slave transmitter recovery from deadlock situation

While operating the USIC channel as an IIC slave transmitter, if the slave runs out of data to transmit before the master receiver issues clock pulses, for example due to an error in the application flow, it ties the SCL infinitely low.

Recommendation

To recover and reinitialize the USIC IIC slave from such a deadlock situation, the following software sequence can be used:

1. Switch the SCL and SDA port functions to be general port inputs for the slave to release the SCL and SDA lines:
 - a) Write 0 to the two affected Pn_IOCRx.PCy bit fields.
2. Flush the FIFO buffer:
 - a) Write 1_B to both USICx_CHy_TRBSCR.FLUSHTB and FLUSHRB bits.
3. Invalidate the internal transmit buffer TBUF:
 - a) Write 10_B to USICx_CHy_FMR.MTDV.
4. Clear all status bits and reinitialize the IIC USIC channel if necessary.
5. Reprogram the Pn_IOCRx.PCy bit fields to select the SCL and SDA port functions again.

At the end of this sequence, the IIC slave is ready to communicate with the IIC master again.