Errata Sheet

Rel. 1.10, 2016-11

Device	**XMC1300**
Marking/Step	**EES-AA, ES-AA, AA**
Package	**PG-TSSOP-16/38, PG-VQFN-24/40**

Overview

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

Table 1 Current User Documentation

Document	Version	Date
XMC1300 Reference Manual	V1.1	Apr 2014
XMC1300 Data Sheet	V1.4	May 2014

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
1.10	2016-11	This Document. Removed Firmware_CM.002 - not applicable for AA devices. Other changes see column “Chg” in the table below.

Table 3 Errata fixed in this step

Errata	Short Description	Change
- none -		

Table 4 Functional Deviations

Functional Deviation	Short Description	XMC1301	XMC1302	Chg	Pg
ACMP_CM.001	Operating range of the Analog Comparator Reference Divider function	X	X		10
ADC_AI.003	Additonal bit to enable ADC function	X	X		10
ADC_AI.004	ADC Calibration Weakness	X	X	Update	10
ADC_AI.008	Wait-for-Read condition for register GLOBRES not detected in continuous auto-scan sequence	X	X		12
ADC_AI.010	ADC Operating Range	X	X		13
ADC_AI.013	Sigma-Delta Loop	X	X		13
ADC_AI.014	Wrong Result of Conversion in Cancel-Inject-Repeat Mode	X	X		13
ADC_AI.015	Sporadic Result Errors when Operated in Low Voltage Range	X	X		14

Table 4 Functional Deviations (cont'd)

Functional Deviation	Short Description	XMC1301	XMC1302	Chg	Pg
ADC_AI.016	No Channel Interrupt in Fast Compare Mode with GLOBRES	X	X		14
BCCU_CM.001	Channel output not switched to passive level when channel is disabled		X		14
BCCU_CM.002	No interrupt generated when software trap is triggered via EVFSR.TPS		X		15
BCCU_CM.003	Channel shadow transfer bit is cleared on wrong clock		X		15
BCCU_CM.004	Dimming engine shadow transfer bit is cleared on wrong clock		X		16
BCCU_CM.005	Disallowed ONCMP-OFFCMP combinations		X		16
BCCU_CM.006	No packer trigger for stable signal if channel is configured for falling edge trigger		X		16
BCCU_CM.007	Shadow process with dithering may not reach target level if follows a bypass shadow process		X		17
BCCU_CM.008	Linear walk starts with a delay after an aborted linear walk		X		17
BCCU_CM.009	Dimming level not immediately changed for first dimming operation		X		17
BCCU_CM.010	Shadow process with dithering may not reach target level if dimming level is previously set to 1-127		X		18

Table 4 Functional Deviations (cont'd)

Functional Deviation	Short Description	XMC1301	XMC1302	Chg	Pg
BCCU_CM.011	Trigger mode 1 cannot be used with trigger delay		X		18
CCU_AI.005	CCU4 and CCU8 External IP clock Usage	X	X		19
CCU_AI.006	Value update not usable in period dither mode	X	X		20
CCU_AI.008	Clock ratio limitation when using MCSS inputs	X	X	New	21
CCU8_AI.002	CC82 Timer of the CCU8x module cannot use the external shadow transfer trigger connected to the POSIFx module	X	X		22
CCU8_AI.003	CCU8 Parity Checker Interrupt Status is cleared automatically by hardware	X	X		24
CCU8_AI.004	CCU8 output PWM glitch when using low side modulation via the Multi Channel Mode	X	X		27
CCU8_AI.006	Timer concatenation does not work when using external count signal	X	X	New	30
CPU_CM.002	Watchpoint PC functions can report false execution	X	X		31
CPU_CM.003	Prefetch faulting instructions can erroneously trigger breakpoints	X	X		33
Firmware_CM.001	User routine _NvmProgVerify stalls the system bus for two to three maximum 10 µs periods	X	X		33
Firmware_CM.004	SSC BSL is not supported	X	X	New	34

Table 4 Functional Deviations (cont'd)

Functional Deviation	Short Description	XMC1301	XMC1302	Chg	Pg
Firmware_CM.005	Last byte of SRAM is not available for ASC BSL	X	X	New	34
Firmware_CM.006	Header resend not supported for incorrect ASC BSL header byte	X	X	New	34
NVM_CM.001	NVM Write access to trigger NVM erase operation must NOT be executed from NVM	X	X		35
NVM_CM.002	Completion of NVM verify-only operations do not trigger NVM interrupt	X	X		36
PORTS_CM.004	Outputs of CCU4, BCCU and ACMP cannot be used to effectively control the pull devices on Pin	X	X		36
POSIF_AI.001	Input Index signal from Rotary Encoder is not decoded when the length is 1/4 of the tick period	X	X		37
SCU_CM.010	Handling of Master Reset via bit RSTCON.MRSTEN	X	X		39
SCU_CM.011	Incomplete Initialisation after a System Reset	X	X		39
SCU_CM.012	Calibrating DCO based on Temperature Sensor	X	X		39
SCU_CM.013	Brownout reset triggered by External Brownout Detector (BDE)	X	X		40
SCU_CM.014	Temperature Sensor User Routines in ROM	X	X		40
SCU_CM.016	Usage of Offset Formulae for DCO Calibration based on Temperature	X	X		40

Table 4 Functional Deviations (cont'd)

Functional Deviation	Short Description	XMC1301	XMC1302	Chg	Pg
SCU_CM.018	Accuracy of Temperature Sensor out of specification	X	X		41
SCU_CM.020	DCO nominal frequencies and accuracy based on Temperature Sensor calibration	X	X		42
USIC_AI.008	SSC delay compensation feature cannot be used	X	X	New	43
USIC_AI.014	No serial transfer possible while running capture mode timer	X	X		43
USIC_AI.017	Clock phase of data shift in SSC slave cannot be changed	X	X		43
USIC_AI.018	Clearing PSR.MSLS bit immediately deasserts the SELOx output signal	X	X		44
WDT_CM.001	No overflow is generated for WUB default value	X	X	New	44

Table 5 Deviations from Electrical- and Timing Specification

AC/DC Deviation	Short Description	XMC1301	XMC1302	Chg	Pg
ADC_AI.P002	DC Switching Level (V_{ODC}) of Out of Range Comparator	X	X		46

Table 6 Application Hints

Hint	Short Description	XMC1301	XMC1302	Chg	Pg
ACMP_CM.H002	Disabling the ORC	X	X	New	47
ADC_AI.H006	Ratio of Module Clock to Converter Clock	X	X		47
ADC_AI.H007	Ratio of Sample Time t_S to SHS Clock f_{SH}	X	X		48
ADC_AI.H009	ADC Operation with internal reference, lower supply voltage range	X	X		49
BCCU_CM.H001	Additional dimming clocks after dimming curve switch		X		49
BCCU_CM.H002	BCCU clocks may not freeze in Suspend Mode		X		49
BCCU_CM.H003	Dimming engine output not cleared upon disabling of dimming engine		X		49
BCCU_CM.H004	Packer threshold (CHCONFIGy.PKTH) accepted values		X		50
BCCU_CM.H005	Enable a dimming engine for global dimming		X		50
Firmware_CM.H001	Switching to high baudrates in enhanced ASC BSL	X	X		50
Firmware_CM.H002	Ensuring correct selection of RxD Pin in ASC Bootstrap Loader	X	X		52
NVM_CM.H001	Adding a wait loop to stand-alone verification sequences	X	X		53

Table 6 Application Hints (cont'd)

Hint	Short Description	XMC1301	XMC1302	Chg	Pg
SCU_CM.H001	Temperature Sensor Functionality	X	X		53
USIC_AI.H004	I2C slave transmitter recovery from deadlock situation	X	X		54

Table 7 Documentation Updates

Hint	Short Description	XMC1301	XMC1302	Chg	Pg
ADC_CM.D001	Definition of trigger bus bits in register OCS is wrong	X	X	New	55
ACMP_CM.D001	Incorrect description of ACMP reference divider function	X	X		55
Firmware_CM.D001	Incorrect specification of length of Chip Variant Identification Number	X	X		56
Firmware_CM.D002	Incorrect specification of value of Status Indicators returned by NVM routines	X	X		56
STARTUP_CM.D001	SSC Bootstrap Loader Identification Byte is documented wrong	X	X	New	57
WDT_CM.D001	Correction to section "Pre-warning Mode"	X	X	New	57

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.003 Additonal bit to enable ADC function

The analog section of ADC is not fully functioning when it is enabled by bit GxARBCFG (x = 0 - 1).ANONS and bit SHSCFG.ANOFF.

Workaround

To enable the analog section of the ADC, at least one of the out-of-range comparators must be enabled in addition to the setup as mentioned above. This is done by setting at least one of bits ENORCx (x = 0 - 7) in register ORCCTRL.

ADC_AI.004 ADC Calibration Weakness

The calibration mechanisms of the ADC show a problem with the offset calibration.This leads to inaccurate result values and, therefore, requires additional actions.

Workaround

Additional actions are recommended for ADC initialization and during operation.

During ADC initialization and before start of calibration, the following sequence is required:

- Enable Analog Converter to normal mode,
  GxARBCFG(x = 0 - 1).ANONS = 0x03
  SHS0_SHSCFG.ANOFF = 0
- Wait until Converter has turned on, SHS0_SHSCFG.ANRDY = 1
- Add approximately 15 µsec for the ADC power to stabilize
- Configure the sample and conversion time

Next, trigger the startup calibration and gain calibration loop:

- Startup Calibration
  a) Initiate start up calibration, GLOBCFG.SUCAL = 1
  b) Disable Post calibration, GLOBCFG.DPCAL0 = 1
  c) Wait until start-up calibration is started, G0ARBCFG.CALS = 1
  d) Clear offset calibration values[1)]
  for 1920 calibration cycles
  while waiting for start-up calibration to finish
  , G0ARBCFG.CAL = 0
  e) Clear again the offset calibration values[1)] before exit.
- Gain calibration workaround loop
  a) Set
  CALMAX to maximum value, SHS0_CALCTR.CALMAX = 3F
  calibration maximum timing by writing $3F100400_H$ to register address $480340BC_H$
  b) Setup group 0 channel for conversion.
  c) Enable post calibration for group 1 and group 0, GLOBCFG.DPCAL1 = GLOBCFG.DPCAL0 = 0
  d) Clear offset calibration values.[1)]
  e) Execute 9 x 2000 dummy conversions and clear offset calibration values[1)] after each conversion.
  f) Clear offset calibration values[1)] while waiting for the post calibration loop to finish, SHS0_SHSCFG.STATE = 0
  g) Reset the configuration used for dummy conversion.

1) Offset calibration values are cleared by writing value 00008000_H to register addresses $480340E0_H$ and $480340E4_H$.

After the end of the gain calibration workaround loop, configure the ADC for user application.

During runtime:

- Since a post calibration cycle is executed automatically after each conversion cycle, it is sufficient to clear offset values[1)] after retrieving a result value.Calibration steps are automatically inserted when no conversions are executed. To avoid miscalibration, ensure that the offset values are cleared[1)] before a lapse of 1024 µs.

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 (GLOBCR.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.010 ADC Operating Range

ADC operation at 3.5 V to 5.5 V is covered by production test. Other range is not yet covered by production test. Gain error may increase at 3.0 V to 3.5 V and 1.8 V to 2.2 V.

Workaround

None.

ADC_AI.013 Sigma-Delta Loop

The sigma-delta loop does not operate as specified and, therefore, cannot be used.

Workaround

None.

ADC_AI.014 Wrong Result of Conversion in Cancel-Inject-Repeat Mode

If a running conversion (A) is aborted by a higher prioritized (injected) conversion (B) on the same group Gx in a time window close to end of sampling of conversion A, the result of conversion B may be considerably wrong.

Workaround

The problematic time frame can be avoided by ensuring a sample time shorter than an arbitration round.

Example:

For a sample time of 100 ns, the arbitration round length t_{ARB} should be programmed to 4 * t_{ADC} (e.g. with DIVD = 0, ARBRND = 0, i.e. t_{ARB} = 4 * (DIVD+1) * t_{ADC}, @ f_{ADC} = 32 MHz).

ADC_AI.015 Sporadic Result Errors when Operated in Low Voltage Range

When the ADC is operated in low voltage range (SHSCFG.AREF = 11_B, internal reference), the result values may be sporadically inaccurate.

Workaround

Attenuate the noise created by these inaccurate results by averaging several result values or using a filter. A median filter is suitable.

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 selected result register and the result of the comparison is stored in the respective bit FCR.

A channel event can be generated when the input becomes higher or lower than the compare value.

In case the global result register GLOBRES is selected, the comparison is executed correctly, the target bit is stored correctly, source events and result events are generated, but a channel event is not generated.

Workaround

If channel events are required, choose a local result register GxRESy for the operation of the fast compare channel.

BCCU_CM.001 Channel output not switched to passive level when channel is disabled

When an active channel is disabled by clearing CHEN.ECHy, the channel output will not go to its passive level as determined by CHOCON.CHyOP. Instead, the channel output will just stay at its last level.

Workaround

The channel intensity must be changed to 0 before disabling the channel.

Pseudocode:

```
CHCONFIGy.LINPRES = 0;
INTSy.TCHINT = 0;
CHSTRCON.CHyS = 1;
while (!CHSTRCON.CHyS) CHSTRCON.CHyS = 1;
CHEN.ECHy = 0;
```

BCCU_CM.002 No interrupt generated when software trap is triggered via EVFSR.TPS

Generating a software TRAP by setting EVFSR.TPS will set TPSF but not TPF. The behaviour is different from a hardware trap because no interrupt will be generated.

Workaround

EVFSR.TPS and EVFSR.TPFS must be set at the same time.

BCCU_CM.003 Channel shadow transfer bit is cleared on wrong clock

CHSTRCON.CHyS is cleared by hardware when the linear walk is complete and the target has been reached. It can only be set again one BCCU_fclk period later (determined by GLOBCLK.FCLK_PS). Write attempts before this period time is up will be ignored.

Workaround

Repeat setting CHSTRCON.CHyS until success

Pseudocode:

```
while (!CHSTRCON.CHyS) CHSTRCON.CHyS = 1;
```

BCCU_CM.004 Dimming engine shadow transfer bit is cleared on wrong clock

DESTRCON.DEzS is cleared by hardware when the dimming process is complete and the target has been reached. It can only be set again one BCCU_dclk period later (determined by GLOBCLK.DCLK_PS). Write attempts before this period time is up will be ignored.

Workaround

Repeat setting DESTRCON.DEzS until success

Pseudocode:

```
while (!DESTRCON.DEzS) DESTRCON.DEzS = 1;
```

BCCU_CM.005 Disallowed ONCMP-OFFCMP combinations

Certain ONCMP-OFFCMP combinations, including the default value, are not allowed. The packer is not functional with these.ONCMP should have a value above 5 or OFFCMP should have a value lesser than 249.

Workaround

Use values in the recommended range for ONCMP and OFFCMP.

BCCU_CM.006 No packer trigger for stable signal if channel is configured for falling edge trigger

When the channel is configured for falling edge trigger, the packer issues a trigger for ON->OFF state transitions. However, no triggers are issued for OFF->OFF and ON->ON state transitions (stable signals). This will cause the round robin to get stuck for Trigger Mode 1 (GLOBCON.TM = 1_B).

Workaround

Disable the packer (CHCONFIGy.PEN = 0_B) and enable the forced trigger (CHCONFIGy.ENFT = 1_B) to achieve the same behavior.

BCCU_CM.007 Shadow process with dithering may not reach target level if follows a bypass shadow process

If the dimming level is previously set to level 1-127 via bypass shadow process (DIMDIV = 0_B) and is followed by a shadow process with dithering active, the target level reached in the second process is not as desired.

Workaround

After the shadow process with dithering, check the dimming level and adjust manually if necessary.

Pseudocode:

```
while(DESTRCON.DEzS==1);
if(DLz.DLEV!=target_level)
{
        CHCONFIGy.DPB=1;
        DLSz.TDLEV=target_level;
        DESTRCON.DEzS=1;
}
```

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.

BCCU_CM.010 Shadow process with dithering may not reach target level if dimming level is previously set to 1-127

The target dimming level may not be reached if a shadow process with dithering active is triggered after a shadow process (with or without dithering) which sets the dimming level to between 1-127.

Workaround

After the shadow process with dithering, check the dimming level and adjust manually if necessary.

Pseudocode:

```
while(DESTRCON.DEzS==1);
if(DLz.DLEV!=target_level)
{
        CHCONFIGy.DPB=1;
        DLSz.TDLEV=target_level;
        DESTRCON.DEzS=1;
}
```

BCCU_CM.011 Trigger mode 1 cannot be used with trigger delay

If trigger mode 1 is selected (GLOBCON.TM = 1_B) with a trigger delay (GLOBCON.TRDEL = 01_B or 10_B), the trigger output is sometimes generated at the incorrect trigger signal. Trigger mode 1 with no delay (GLOBCON.TRDEL = 00_B or 11_B) is still functional.

Workaround

None

CCU_AI.005 CCU4 and CCU8 External IP clock Usage

Each CCU4/CCU8 module offers the possibility of selecting an external signal to be used as the master clock for every timer inside the module Figure 1. External signal in this context is understood as a signal connected to other module/IP or connected to the device ports.

The user has the possibility after selecting what is the clock for the module (external signal or the clock provided by the system), to also select if this clock needs to be divided. The division ratios start from 1 (no frequency division) up to 32768 (where the selected timer uses a frequency of the selected clock divided by 32768).

This division is selected by the PSIV field inside of the CC4yPSC/CC8yPSC register. Notice that each Timer Slice (CC4y/CC8y) have a specific PSIV field, which means that each timer can operate in a different frequency.

Currently is only possible to use an external signal as Timer Clock when a division ratio of 2 or higher is selected. When no division is selected (divided by 1), the external signal cannot be used.

The user must program the PSIV field of each Timer Slice with a value different from 0000_B - minimum division value is /2.

This is only applicable if the Module Clock provided by the system (the normal default configuration and use case scenario) is not being used. In the case that the normal clock configured and programmed at system level is being used, there is not any type of constraints.

One should not also confuse the usage of an external signal as clock for the module with the usage of an external signal for counting. These two features are completely unrelated and there are not any dependencies between both.

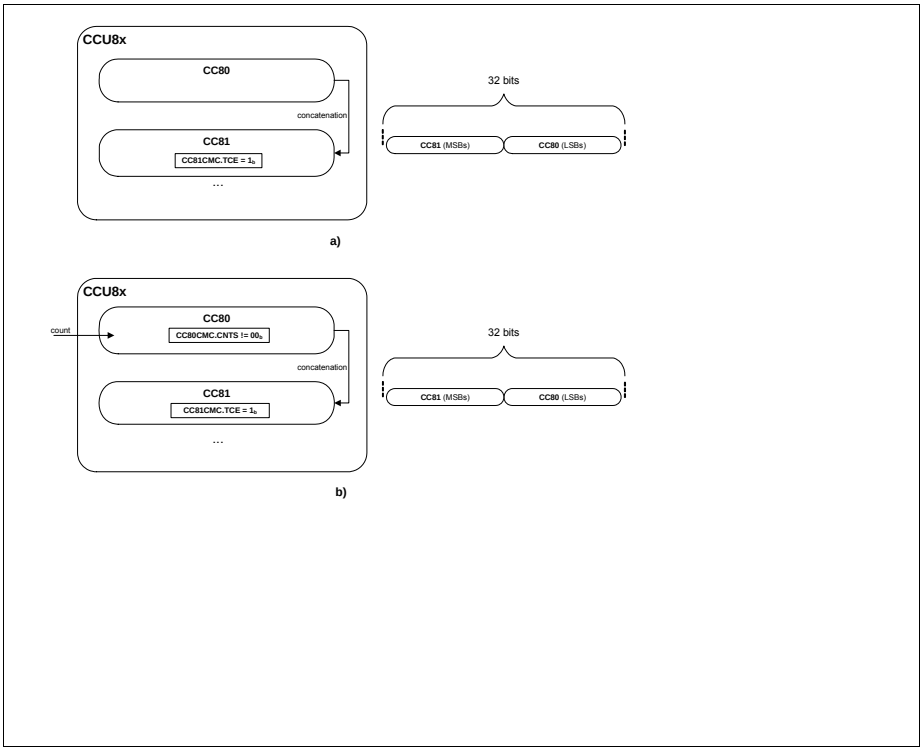

Figure 1 Clock Selection Diagram for CCU4/CCU8

Workaround

None.

CCU_AI.006 Value update not usable in period dither mode

Each CCU4/CCU8 timer gives the possibility of enabling a dither function, that can be applied to the duty cycle and/or period. The duty cycle dither is done to increase the resolution of the PWM duty cycle over time. The period dither is done to increase the resolution of the PWM switching frequency over time.

Each of the dither configurations is set via the DITHE field:

- DITHE = 00_B - dither disabled

- DITHE = 01_B - dither applied to the duty-cycle (compare value)
- DITHE = 10_B - dither applied to the period (period value)
- DITHE = 11_B - dither applied to the duty-cycle and period (compare an period value)

Whenever the dither function is applied to the period (DITHE = 10_B or DITHE = 11_B) and an update of the period value is done via a shadow transfer, the timer can enter a stuck-at condition (stuck at 0).

Implication

Period value update via shadow transfer cannot be used if dither function is applied to the period (DITHE programmed to 10_B or 11_B).

Workaround

None

CCU_AI.008 Clock ratio limitation when using MCSS inputs

The MCSS input signals of CCU8 and CCU4 units are erroneously sampled with the AHB bus clock f_{PERIPH} instead of the module clock f_{CCU}.

Implication

If the f_{PERIPH} and f_{CCU} frequencies are programmed to a ratio different from 1:1 then the MCSS signals running from POSIF to CCU4/CCU8 are not correctly sampled by CCU8/CCU4.

This can for example affect brushless DC motor drive applications when a clock ratio different from 1:1 is required.

Workaround

None

CCU8_AI.002 CC82 Timer of the CCU8x module cannot use the external shadow transfer trigger connected to the POSIFx module

Each CCU8 Module Slice contains 4 identical timers (CC80, CC81, CC82 and CC83). There is the possiblity of updating the values controlling the duty cycle, period, output passive level, dither and floating prescaler on-the-fly of each and every timer, with a SW request. The update request of these values can also be done via an external trigger that is connected to the POSIFx module Figure 1. An update action of any of these values is named as “shadow transfer”.

The signal between the POSIFx and CCU8x module is used to handshake a concurrent update between several registers, contained in the two modules. The output signal of the POSIFx is named as POSIFx.OUT6 while the input signal on the CCU8x side is named as CCU8x.MCSS.

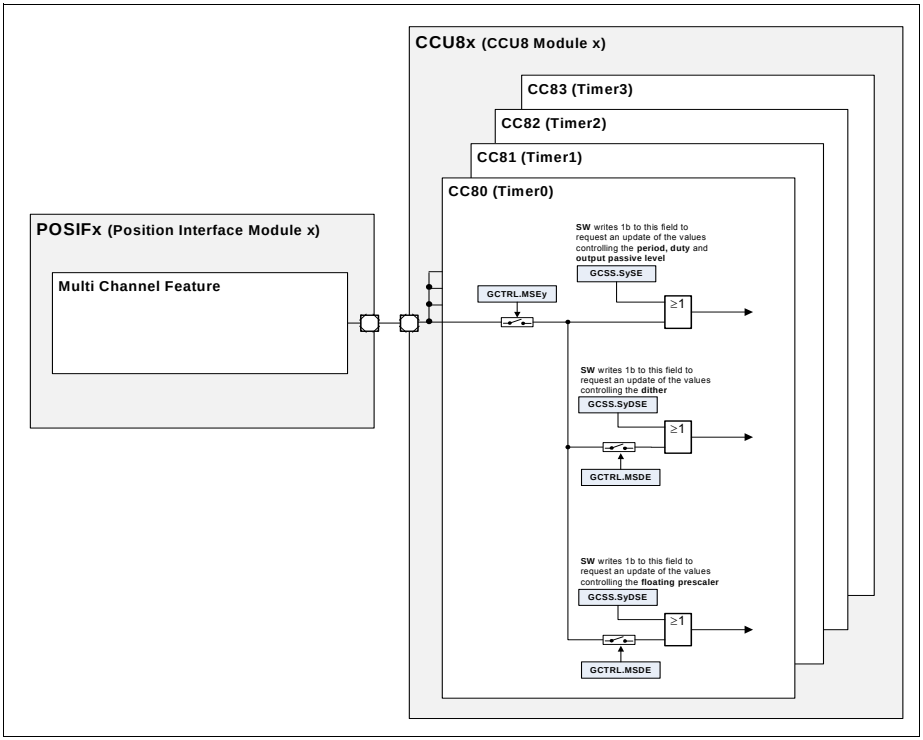

Figure 2 Value update trigger connection between CCU8x and POSIFx

On Figure 2, we see an example how this trigger is used to update at the same time teh duty cycle value of a timer inside the CCU8x and the multi channel pattern inside the POSIFx (the multi channel pattern can affect the CCU8x timer outputs therefore a synchronous update of all the values solves possible output glitches on the generated PWM signals).

On Figure 2, the SW has updated the next values for the duty cycle on a CCU8x Timer (it can be also for the period, output passive level and clock prescaler). After that it updates also the next value of the multi channel pattern inside the POSIFx module. After that, the POSIF reaches an internal state (dictated by specific conditions) where an update of the values is needed. it generates a trigger signal to the CCU8x Timer to signalize that an update of the duty cycle value needs to be done. After that timeframe, the POSIF waits for the handshake trigger of the CCU8x Timer to indicate that an update is going to be performed. At this specific time, both the values of the CCU8x Timer and POSIF are update completely synchronous.

This feature cannot be used with the Timer2 (defined as CC82 in the documentation) of the CCU8x module(s) (more than one CCU8 module can be contained on a specific device).

All the other 3 Timers (defined as CC80, CC81, CC83) inside the CCU8x moduel are not affected by this issue.

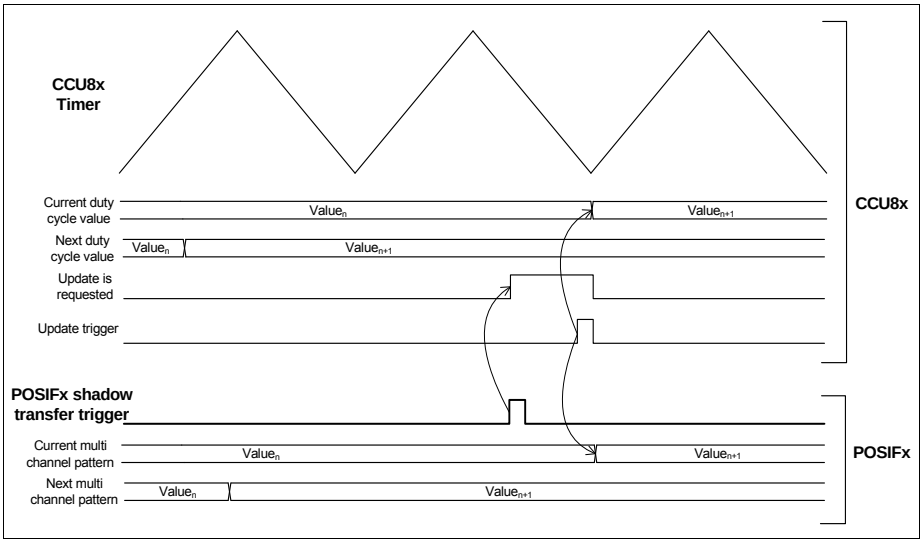

Figure 3 Value update handshake between CCU8x and POSIFx

Workaround

None

CCU8_AI.003 CCU8 Parity Checker Interrupt Status is cleared automatically by hardware

Each CCU8 Module Timer has an associated interrupt status register. This Status register, CC8yINTS, keeps the information about which interrupt source triggered an interrupt. The status of this interrupt source can only be cleared by software. This is an advantage because the user can configure multiple interrupt sources to the same interrupt line and in each triggered interrupt routine, it reads back the status register to know which was the origin of the interrupt.

Each CCU8 module also contains a function called Parity Checker. This Parity Checker function, crosschecks the output of a XOR structure versus an input signal, as seen in Figure 1.

When using the parity checker function, the associated status bitfield, is cleared automatically by hardware in the next PWM cycle whenever an error is not present.

This means that if in the previous PWM cycle an error was detected and one interrupt was triggered, the software needs to read back the status register before the end of the immediately next PWM cycle.

This is indeed only necessary if multiple interrupt sources are ORed together in the same interrupt line. If this is not the case and the parity checker error source is the only one associated with an interrupt line, then there is no need to read back the status information. This is due to the fact, that only one action can be triggered in the software routine, the one linked with the parity checker error.

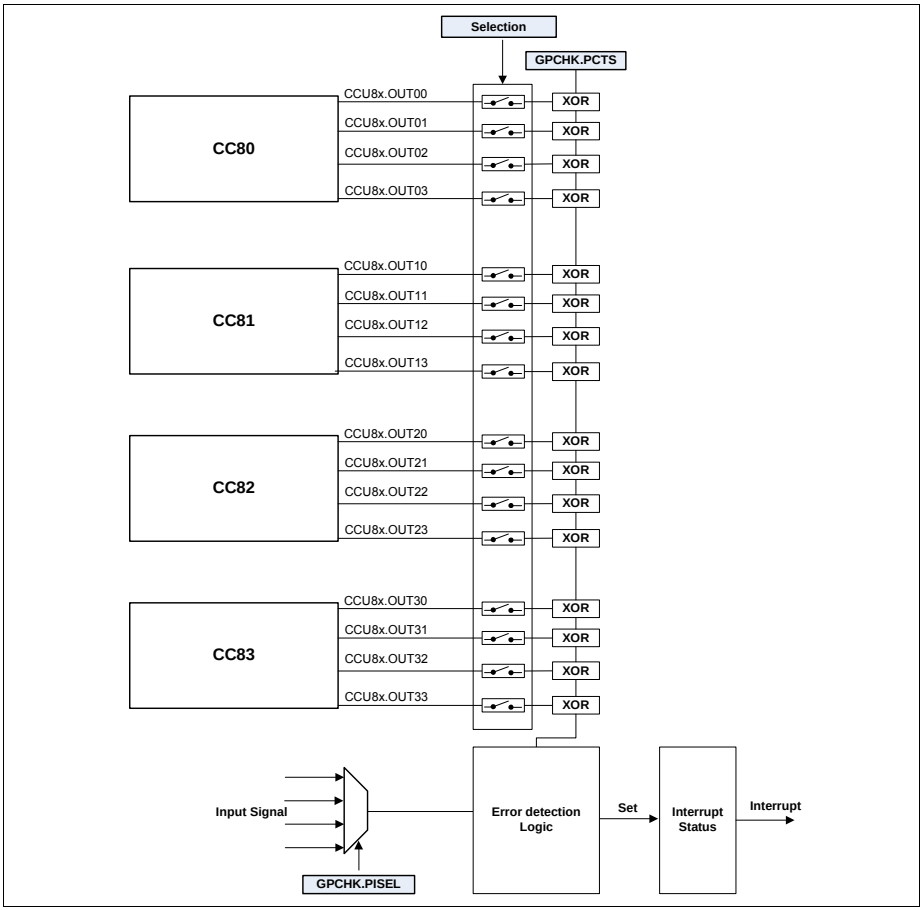

Figure 4 Parity Checker diagram

Workaround

Not ORing the Parity Checker error interrupt with any other interrupt source. With this approach, the software does not need to read back the status information to understand what was the origin of the interrupt - because there is only one source.

CCU8_AI.004 CCU8 output PWM glitch when using low side modulation via the Multi Channel Mode

Each CCU8 Timer Slice can be configured to use the Multi Channel Mode - this is done by setting the CC8yTC.MCME1 and/or CC8yTC.MCME2 bit fields to 1_B. Each bit field enables the multi channel mode for the associated compare channel of the CCU8 Timer Slice (each CCU8 Timer Slice has two compare channels that are able to generate each a complementary pair of PWM outputs).

After enabled, the Multi Channel mode is then controlled by several input signals, one signal per output. Whenever an input is active, the specific PWM output is set to passive level - Figure 1.

The Multi Channel mode is normally used to modulate in parallel several PWM outputs (a complete CCU8 - up to 16 PWM signals can be modulated in parallel).

A normal use case is the parallel control of the PWM output for BLDC motor control. In Figure 2, we can see the Multi Channel Pattern being updated synchronously to the PWM signals. Whenever a multi channel input is active (in this case 0), the specific output is set into passive level (the level in which the external switch is OFF).

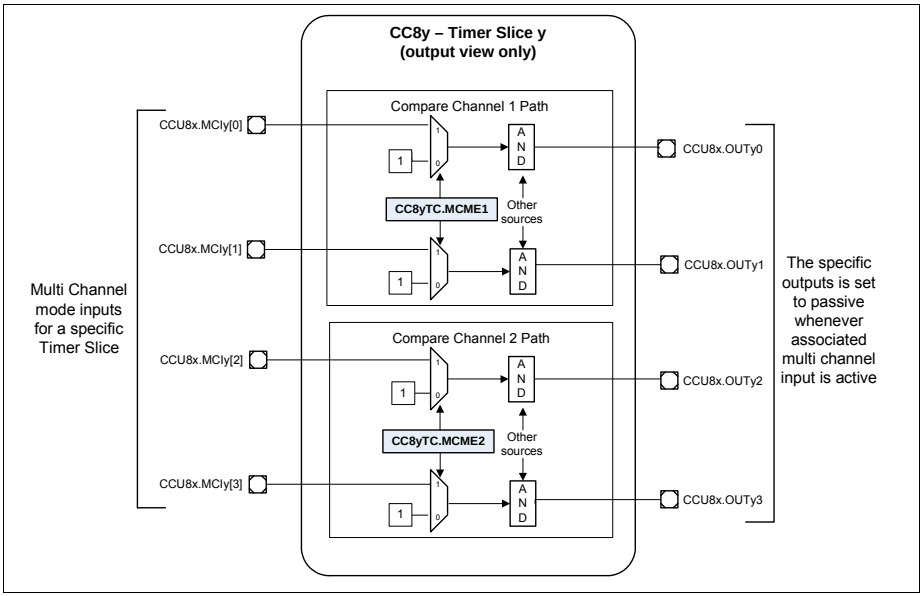

Figure 5 Multi Channel Mode diagram

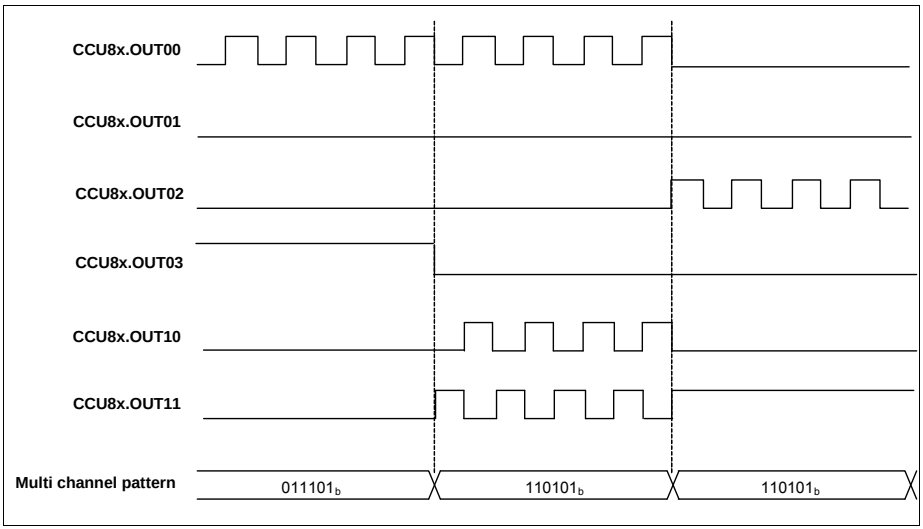

Figure 6 Multi Channel Mode applied to several CCU8 outputs

A glitch is present at the PWM outputs whenever the dead time of the specific compare channel is enabled - CC8yDTC.DTE1 and/or CC8yDTC.DTE2 set to 1_B (each compare channel has a separate dead time function) - and the specific multi channel pattern for the channel is 01_B or 10_B.

This glitch is not present if the specific timer slice is configure in symmetric edge aligned mode - CC8yTC.TCM = 0_B and CC8yCHC.ASE = 0_B.

This glitch only affects the PWM output that is linked to the inverting ST path of each compare channel (non inverting outputs are not affected).

The effect of this glitch can be seen in Figure 3. The duration of the PWM glitch has the same length has the dead time value programmed into the CC8yDC1R.DT1F field (for compare channel 1) or into the CC8yDC1R.DT2F.

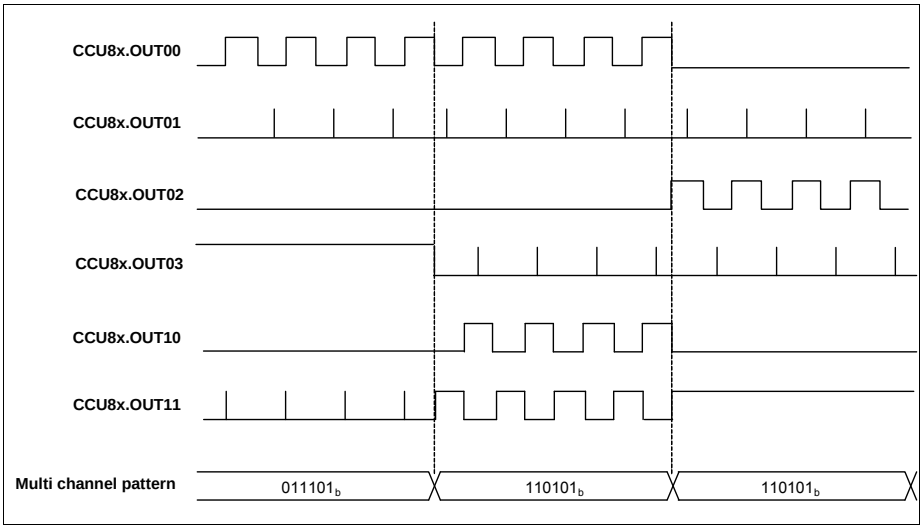

Figure 7 PWM output glitch

Workaround

To avoid the glitch on the inverting path of the PWM output, one can disable the dead time function before the Multi Channel Pattern is set to 01_B or 10_B. Disabling the dead time of the inverting PWM output can be done by setting:

```
CC8yDTC.DCEN2 = 0
```

```
CC8yDTC.DCEN4 = 0
```

The dead time needs to be re enabled, before the complementary outputs become modulated at the same time:

```
CC8yDTC.DCEN2 = 1
CC8yDTC.DCEN4 = 1
```

CCU8_AI.006 Timer concatenation does not work when using external count signal

Each CCU8 peripheral contains four sixteen bit timers. It is possible nevertheless to concatenate multiple timers to achieve a timer/counter with 32, 48 or 64 bits. To enable the concatenation feature, the CC8yCMC.TCE bitfield needs to be set to 1_B - Figure 1 a), where CCU8x represents a CCU8 peripheral instance x, and CC80 and CC81, represents timer 0 and timer 1 respectively (please notice that CC80 and CC81 are just used for simplicity, meaning that this function can be used also with the other timers inside CCU8x).

It is also possible to use an external signal as a count trigger. This means that when using an external count signal, the LSB timer is incremented each time that a transition on this external signal occurs - Figure 1 b).

When an external count signal is enabled - by programming the CC8yCMC.CNTS with 01_B, 10_B or 11_B - the concatenation function does not work. One cannot use in parallel the timer concatenation and external count signal features.

Note: On Figure 1, the count signal is used in CCU80 because this timer represents the LSBs. While the count signal could be enabled in the MSB timer (CC81), this does not make sense when the timers are concatenate - because the count should be used to increment the LSB timer. The LSB timer will then in each wrap around, increment the MSB timer.

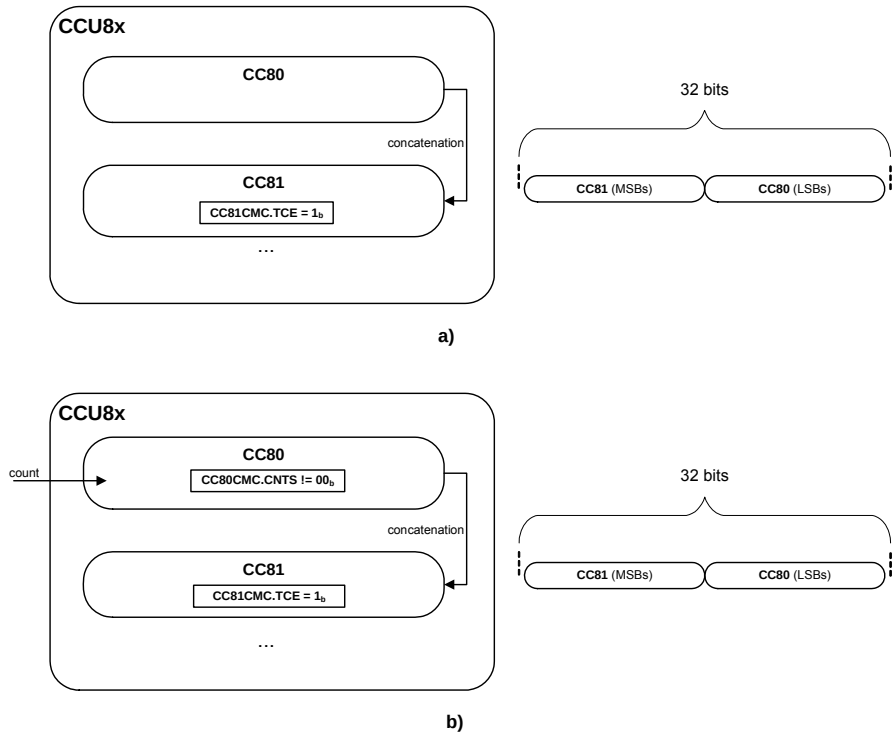

Figure 8 CCU8x concatenation feature resource configuration - a) without external count function; b) with external count function

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.001 User routine _NvmProgVerify stalls the system bus for two to three maximum 10 µs periods

The user routine “Erase, Program and Verify Flash Page” (_NvmProgVerify) in the Boot ROM stalls the system bus for two to three periods, the duration of each period being maximum 10 µs. The bus stall is the result of accessing the NVM while NVM is busy.

During these periods when the bus is stalled, any interrupts generated will be delayed until the bus becomes available again. This is the case even for interrupts that have their handlers located in the SRAM, since all memory accesses have to go through the system bus.

Workaround

None.

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 8** is expected from the host.

An incorrect Header Byte received leads to a hang-up of the device.

Table 8 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.

NVM_CM.001 NVM Write access to trigger NVM erase operation must NOT be executed from NVM

When the NVM write access to trigger an NVM erase operation is executed from NVM, the erase operation is not always executed.

Implications

This issue only affects the NVM operation ERASE. The remaining NVM operations WRITE and VERIFY are not affected.

Workaround

When implementing the Low-Level Programming Routines, the programmer has to take care that the write access to the NVM that is triggering the ERASE operation is not executed from NVM.

It is recommended to use always the NVM user routines provided in the ROM, especially for NVM erase.

NVM_CM.002 Completion of NVM verify-only operations do not trigger NVM interrupt

The completion of either one-shot or continuous verify-only operation (NVMPROG.ACTION = $D0_H$ or $E0_H$ respectively) does not trigger the NVM interrupt, contrary to specifications.

Implications

The NVM interrupt cannot be used to detect for the end of verify-only operations.

Workaround

To detect for the end of verify-only operations, poll the register bit NVMSTATUS.BUSY to be 0 after the specific verify-only operation has started.

PORTS_CM.004 Outputs of CCU4, BCCU and ACMP cannot be used to effectively control the pull devices on Pin

The outputs of BCCU0.OUTx, CCU40.OUTx and ACMPx.OUT can be used to control the internal pull devices via the direct hardware control in the PORTS module.

The intended behaviour is:

- When output is `1`, pull-up device is enable and pull-down device is disable
- When output is `0`, pull-up device is disable and pull-down device is enable

The actual behaviour is:

- When output is `1`, pull-up device is enable and pull-down device is enable
- When output is `0`, pull-up device is disable and pull-down device is disable

Workaround

None

POSIF_AI.001 Input Index signal from Rotary Encoder is not decoded when the length is 1/4 of the tick period

Each POSIF module can be used as an input interface for a Rotary Encoder. It is possible to configure the POSIF module to decode 3 different signals: Phase A, Phase B (these two signals are 90° out of phase) and Index. The index signal is normally understood as the marker for the zero position of the motor Figure 1.

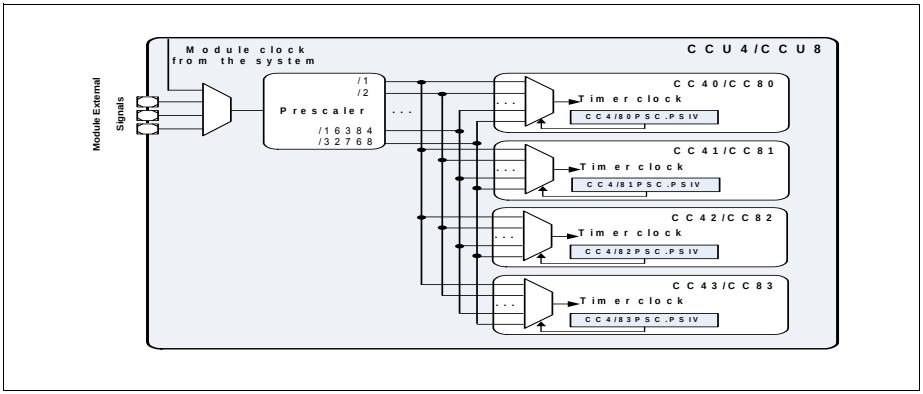

Figure 9 Rotary Encoder outputs - Phase A, Phase B and Index

There are several types of Rotary Encoder when it comes to length of the index signal:

- length equal or bigger than 1 tick period
- length equal or bigger than 1/2 tick period
- length equal or bigger than 1/4 tick period

When the index signal is smaller than 1/2 of the tick period, the POSIF module is not able to decode this signal properly, Figure 2 - notice that the reference edge of the index generation in this figure is the falling of Phase B, nevertheless this is an example and depending on the encoder type, this edge may be one of the other three.

Due to this fact it is not possible to use the POSIF to decode these type of signals (index with duration below 1/2 of the tick period).

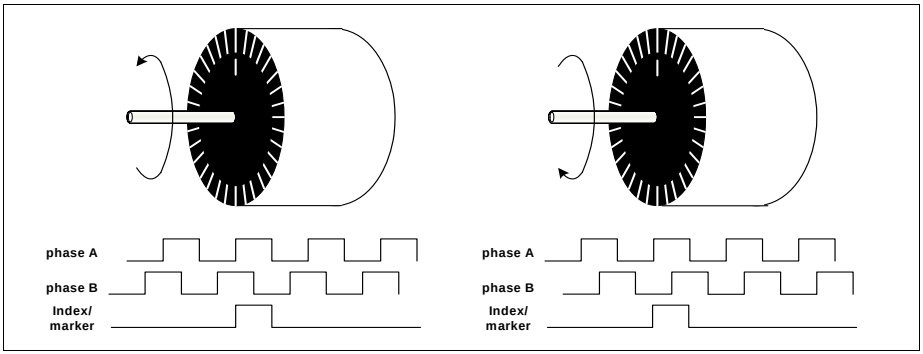

Figure 10 Different index signal types

Workaround

To make usage of the Index signal, when the length of this signal is less than 1/2 of the tick period, one should connect it directly to the specific counter/timer. This connection should be done at port level of the device (e.g. connecting the device port to the specific Timer/Counter(s)), Figure 3.

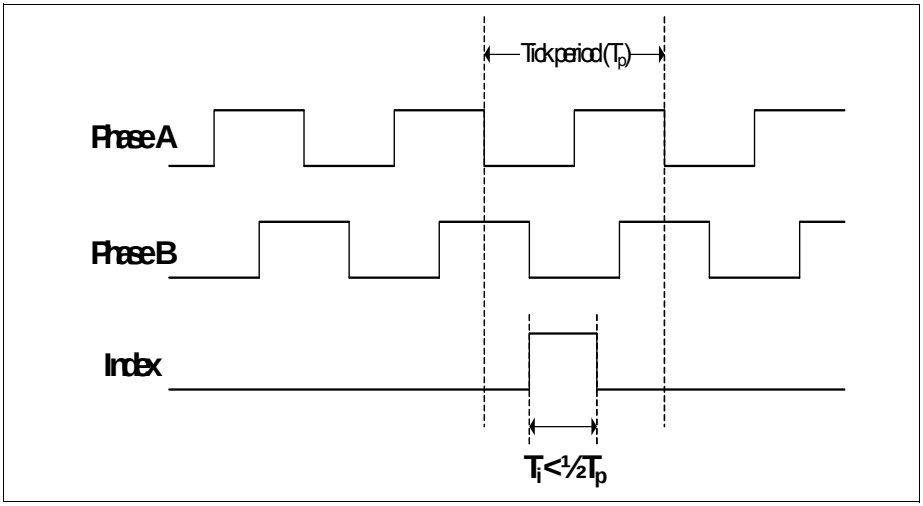

Figure 11 Index usage workaround - a) Non working solution; b) Working solution

SCU_CM.010 Handling of Master Reset via bit RSTCON.MRSTEN

The reset initialisation sequence is incomplete when a Master Reset via bit RSTCON.MRSTEN is triggered after a System Reset while some RSTSTAT.RSTSTAT bit(s) indicating System reset - one or more out of bits [9:2] - is still set.

Workaround

Clear the reset status bits in RSTSTAT.RSTSTAT by setting bit RSTCLR.RSCLR to 1 before triggering the Master Reset.

SCU_CM.011 Incomplete Initialisation after a System Reset

The reset initialisation is incomplete when a System Reset is triggered on devices with Firmware version : $FFFFFFFF_H$. The Firmware version is stored in Flash Configuration Sector 0 (CS0), address $10000FEC_H$.

The issue is solved for devices with a different Firmware version than $FFFFFFFF_H$.

Workaround

When a System Reset happens, it is recommended to trigger the Master Reset via bit RSTCON.MRSTEN after clearing the reset status bits in RSTSTAT.RSTSTAT via bit RSTCLR.RSCLR.

SCU_CM.012 Calibrating DCO based on Temperature Sensor

The function of calibrating DCO based on temperature is not supported in EES and ES samples.

Workaround

None.

SCU_CM.013 Brownout reset triggered by External Brownout Detector (BDE)

Samples with the following marking and Firmware version does not support the BDE brownout detection.

- Package marking of GE247, GE248 or GE249
- Firmware version : $FFFFFFFF_H$ (stored in CS0, address $10000FEC_H$)

The brownout reset may not be triggered when V_{DDP} drops below the V_{DDP} brownout reset voltage.

Workaround

None.

SCU_CM.014 Temperature Sensor User Routines in ROM

The Temperature sensor user routines in ROM cannot be used.

Workaround

Library functions are available and the details of these functions can be found in the Temperature Sensor device guide.

SCU_CM.016 Usage of Offset Formulae for DCO Calibration based on Temperature

In the productive device, DCO1 can be calibrated based on the measured temperature using the temperature sensor(TSE). The offset value for the calibration can be obtained based on the formulae below. The 4 constants are stored in the flash configuration page, where constant d and e may have the values of 0.

(1)

$$OFFSET[steps] = b + \frac{(a-b)(c-d)}{(e-d)}$$

where :

OFFSET value is range from 0 to 8

c is the measured temperature [°C]

a is constant DCO_ADJLO_T2

b is constant DCO_ADJLO_T1

d is constant ANA_TSE_T1

e is constant ANA_TSE_T2

Workaround

If constant d is 0, set d to 25 in the formulae above. If constant e is 0, set e to 115, respectively.

SCU_CM.018 Accuracy of Temperature Sensor out of specification

The temperature sensor accuracy parameter T_{TSAL}, does not fall within the defined limits for the corresponding test conditions in the Temperature Sensor Characteristics table in XMC1000 family Data Sheet V1.4. The deviation of the accuracy is specified in **Table 9**.

Note: The abovementioned deviation does not affect the functionality of the DCO1 calibration based on temperature sensor.

Table 9 Temperature Sensor Characteristics

Parameter	Parameter Symbol	Values			Unit	Test conditions
		Min	Typ.	Max		
Sensor Accuracy	T_{TSAL} CC	-6	–	6	°C	T_J = 25°C, T_J = 70°C, T_J = 115°C
		-10	–	10	°C	T_J = 0°C
		–	+/-12	–	°C	T_J = -25°C
		–	+/-20	–	°C	T_J = -40°C

Workaround

None.

SCU_CM.020 DCO nominal frequencies and accuracy based on Temperature Sensor calibration

The accuracy of DCO1 based on temperature sensor calibration parameter Δf_{LTT} of the 64MHz DCO1 Characteristics table in XMC1000 family Data Sheet V1.4 is not valid.
The min and max limits for f_{NOM} of DCO1 and DCO2 under nominal conditions after trimming are not valid. These limits are defined by the specified accuracy parameter over temperature Δf_{LT}.

Workaround

To improve the accuracy of the DCO1 oscillator, refer to XMC1000 Oscillator Handling Application Note.

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.

WDT_CM.001 No overflow is generated for WUB default value

The Window Watchdog Timer (WDT) does not generate an overflow event if the default counter value $FFFFFFFF_H$ is used in register WUB.

Implications

Without an timer overflow no reset or pre-warning is requested. For other WUB values the WDT operates correctly and a reset or pre-warning is requested upon WDT overflow.

Workaround

Do not use $FFFFFFFF_H$ as counter value.

3 Deviations from Electrical- and Timing Specification

The errata in this section describe deviations from the documented electrical- and timing specifications.

ADC_AI.P002 DC Switching Level (V_{ODC}) of Out of Range Comparator

The DC switching level, V_{ODC}, of the Out of Range Comparator (ORC) is not within the range. It has a minimum value of 30 mV instead of 60 mV and a maximum value of 300 mV instead of 120 mV.

Workaround

None

4 Application Hints

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

ACMP_CM.H002 Disabling the ORC

When the ORC 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.

ADC_AI.H006 Ratio of Module Clock to Converter Clock

For back-to-back conversions, the ratio between the module clock f_{ADC} and the converter clock f_{SH} must meet the limits listed in **Table 10**.

Otherwise, when the internal bus clock $f_{ADC} = f_{MCLK}$ is too slow in relation to the converter clock f_{SH}, the internal result buffer may be overwritten with the result of the next conversion c_2 before the result of the previous conversion c_1 has been transferred to the specified result register.

Table 10 VADC: Ratio of Module Clock to Converter Clock

Conversion Type	f_{ADC} / f_{SH} (min.)	Example for $f_{SH} = f_{CONV}$ = 32 MHz (SHS0_SHSCFG.DIVS = 0)
10-bit Fast Compare Mode (bitfield CMS / CME = 101_B)	3/7	$f_{ADC} = f_{MCLK}$ > 13.72 MHz
Other Conversion Modes (8/10/12-bit)	1/3	$f_{ADC} = f_{MCLK}$ > 10.67 MHz

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

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

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

In this case, the following relation must be fulfilled:

- [(2+STC) x (DIVA+1)] / (DIVS+1) ≥ 4 x t_{CONV}/t_{ADC} = 4 x 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.H009 ADC Operation with internal reference, lower supply voltage range

If the internal reference is used in the lower voltage range, write value $0C_H$ to the second byte of register address $480340BC_H$.

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.H002 BCCU clocks may not freeze in Suspend Mode

Only the clock dividers to FCLK, BCLK and DCLK are frozen in suspend mode. If the divider is frozen in the state in which its clock is enabled, the clock will toggle with the frequency of the BCCU input clock.

BCCU_CM.H003 Dimming engine output not cleared upon disabling of dimming engine

The dimming engine output does not get cleared upon disable. As a result, when the dimming engine is re-enabled, the output is at the level before the dimming engine was disabled.

Before disabling dimming engine, user is recommended to dim to desired level.

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).

Firmware_CM.H001 Switching to high baudrates in enhanced ASC BSL

The ASC Bootstrap Loader allows the user to switch to baudrates higher than the initial baudrate when the communication is established for faster downloading of code/data.

With the current implementation (refer to the “Bootstrap Loaders and User Routines” chapter in Reference Manual) the host device (e.g. a PC) may have problem to switch the baudrate fast enough after sending the request (BSL_STEP as of **Figure 12**) and is not able to receive the device acknowledge (BSL_BR_OK) correctly with the changed ASC channel speed. If this happens, the host will get some error condition - wrong response, start bit not detected, etc. In such a case the host has to ignore the error and send the trailer Byte (BSL_BR_OK) with the new baudrate. The correctness of the communication speed settings will be then decided by the host upon the response from the device after sending the length of code for downloading (refer to **Figure 13**).

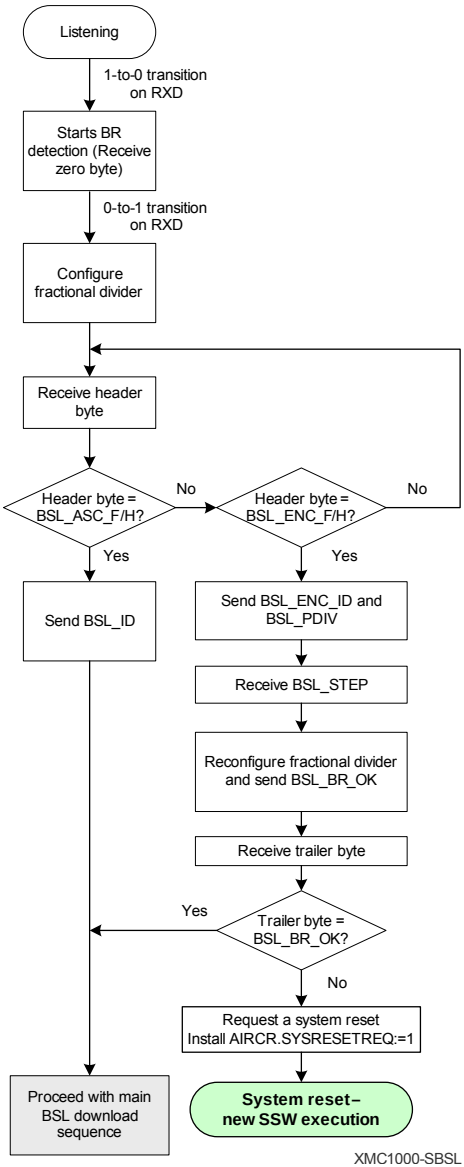

Figure 12 Baud Rate configuration sequence during ASC BSL entry

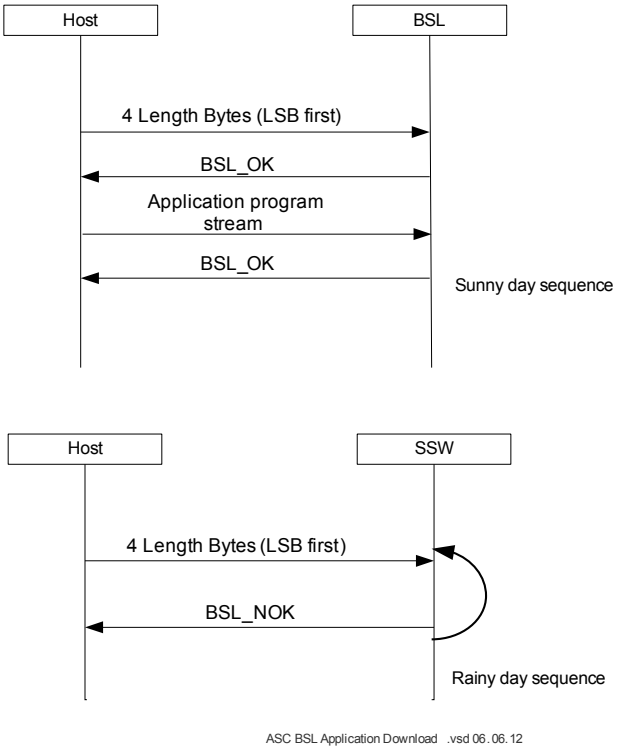

Figure 13 Standard ASC BSL: Application download protocol

Firmware_CM.H002 Ensuring correct selection of RxD Pin in ASC Bootstrap Loader

To provide flexible usage in application, USIC0 channel 0 or 1 are both checked automatically as ASC Bootstrap Loader channel. To prevent possible misidentification of an ASC BSL on the wrong RxD pin, the application must ensure that only the intended pin is activated.

For example, having a capacitor on the pin of an unintended ASC BSL channel, may result in a ramping signal and false detection as the selected ASC BSL channel. Connecting a capacitor to P0.14 when P1.3 is the intended channel,

or to P1.3 when P0.14 is the intended channel, must be avoided when using the ASC Bootstrap Loader.

NVM_CM.H001 Adding a wait loop to stand-alone verification sequences

When a hardread level (NVMCONF.HRLEV = 01_B or 10_B) is selected for a stand-alone verification sequence (NVMPROG.ACTION.VERIFY = 11_B), memory reads from the cell array and register write accesses should be avoided during the transition from VerifyWait to RIdleV state for up to 10 µs, else a bus stall will occur. The NVMSTATUS.BUSY bit remains cleared during this time.

Therefore, it is recommended to insert a wait loop of 10 µs following the completion of the verify sequence, before any write access to SFRs or read/write access to cell array.

Alternatively, if the verify operation is intended following a write operation, it is recommended to use the write operation with automatic verify (NVMPROG.ACTION = 51_H or 61_H), instead of the stand-alone write and verify operations. In this case, the BUSY bit always indicate the actual NVM status and no wait loop will be necessary.

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 a master-issued stop condition, 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.

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

5 Documentation Updates

The errata in this section contain updates to or completions of the user documentation. These updates are subject to be taken over into upcoming user documentation releases.

ADC_CM.D001 Definition of trigger bus bits in register OCS is wrong

The definition of register OCS contains bit fields controlling the OCDS trigger bus (OTBG). This trigger bus is not available in the product.

Correction

The following bit fields of register OCS are invalid:

- TGS
- TGB
- TG_P

The definition is changed to field “0”, type “r”, description “Reserved, write 0, read as 0”.

ACMP_CM.D001 Incorrect description of ACMP reference divider function

In the reference manual v1.1, the description of the analog comparator reference divider function is not correct. The “Analog Comparator Reference Divider function” diagram and ANACMP0 register indicate that ACMP1.INP is connected to ACMP0.INN when bit ANACMP0.ACMP0_SEL is set to 1.

Documentation Update

When bit ANACMP0.ACMP0_SEL is set to 1, ACMP1.INP is connected to ACMP0.INP instead of ACMP0.INN.

Firmware_CM.D001 Incorrect specification of length of Chip Variant Identification Number

In Flash data for SSW and user SW in XMC1300 Table of Reference Manual v1.1, the length of Chip Variant Identification is incorrectly specified as 28B starting from 1000'0F04$_H$.

Documentation Update

The length of Chip Variant Identification should be corrected as 24B starting from 1000'0F04$_H$.

Firmware_CM.D002 Incorrect specification of value of Status Indicators returned by NVM routines

These status indicators values returned by NVM routines in XMC1300 ROM Table of Reference Manual v1.1 are incorrectly specified.

Table 11 Status indicators returned by NVM routines in XMC1300 ROM

Status Indicator		Description
Symbolic name	**Value**	
NVM_E_DST_AREA_EXCEEDED	80010005$_H$	Destination data is not (completely) located in NVM
NVM_E_DST_ALIGNMENT	80010006$_H$	Destination data is not properly aligned
NVM_E_NVM_FAIL	80010009$_H$	NVM module can not be physically accessed
NVM_E_VERIFY	80010010$_H$	Verification of the written page not successful

Documentation Update

The values of the status indicators should be corrected as per below.

Table 12 Status indicators returned by NVM routines in XMC1300 ROM

Status Indicator		Description
Symbolic name	**Value**	
NVM_E_NVM_FAIL	80010005_H	NVM module can not be physically accessed
NVM_E_VERIFY	80010006_H	Verification of the written page not successful
NVM_E_DST_AREA_EXCEEDED	80010009_H	Destination data is not (completely) located in NVM
NVM_E_DST_ALIGNMENT	80010010_H	Destination data is not properly aligned

STARTUP_CM.D001 SSC Bootstrap Loader Identification Byte is documented wrong

Within chapter “SSC Bootstrap Loader” the “Identification Byte” is documented wrong as “$D5_H$”.

The correct value is “$5D_H$”.

WDT_CM.D001 Correction to section "Pre-warning Mode"

Section “Pre-warning Mode” of WDT chapter in the Reference Manual states the following:

"… The alarm status is shown via register WDTSTS and can be cleared via register WDTCLR. A clear of the alarm status will bring the WDT back to normal state. The alarm signal is routed as request to the SCU, where it can be promoted to NMI. …"

Correction

The statement "A clear of the alarm status will bring the WDT back to normal state" is wrong.

A clear of the alarm status bit via write to WDTCLR.ALMC will clear only the bit WDSTSTS.ALMS.

To transfer the WDT back to the normal state a WDT service request is required.