Errata Sheet

Rel. 1.3, 2015-06

Device	**XMC1100**
Marking/Step	**EES-AB, ES-AB, AB**
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
XMC1100 Reference Manual AB-step	V1.2	Nov 2014
XMC1100 Data Sheet AB-step	V1.6	Apr 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
1.3	2015-06	Renamed and updated CCU4_AI.002 to **CCU_AI.006** in **Table 4**; Added **SCU_CM.D001** in **Table 6**

Table 3 Errata fixed in this step

Errata	Short Description	Change
ADC_AI.003	Additonal bit to enable ADC function	Fixed
ADC_AI.004	ADC Calibration Weakness	Fixed
ADC_AI.010	ADC Operating Range	Fixed
ADC_AI.013	Sigma-Delta Loop	Fixed
ADC_AI.015	Sporadic Result Errors when Operated in Low Voltage Range	Fixed
Firmware_CM.001	User routine _NvmProgVerify stalls the system bus for two to three maximum 10 µs periods	Fixed
PORTS_CM.004	Outputs of CCU4, BCCU and ACMP cannot be used to effectively control the pull devices on Pin	Fixed
SCU_CM.010	Handling of Master Reset via bit RSTCON.MRSTEN	Fixed
SCU_CM.011	Incomplete Initialisation after a System Reset	Fixed
SCU_CM.012	Calibrating DCO based on Temperature Sensor	Removed
SCU_CM.013	Brownout reset triggered by External Brownout Detector (BDE)	Fixed
SCU_CM.014	Temperature Sensor User Routines in ROM	Fixed

Table 3 Errata fixed in this step (cont'd)

Errata	Short Description	Change
SCU_CM.016	Usage of Offset Formulae for DCO Calibration based on Temperature	Fixed
NVM_CM.H001	Adding a wait loop to stand-alone verification sequences	Fixed
Firmware_CM.H001	Switching to high baudrates in enhanced ASC BSL	Fixed

Table 4 Functional Deviations

Functional Deviation	Short Description	XMC1100	Chg	Pg
ADC_AI.008	**Wait-for-Read condition for register GLOBRES not detected in continuous auto-scan sequence**	X		**7**
ADC_AI.016	**No Channel Interrupt in Fast Compare Mode with GLOBRES**	X		**7**
CCU_AI.005	**CCU4 and CCU8 External IP clock Usage**	X		**8**
CCU_AI.006	**Value update not usable in period dither mode**	X	Updated	**9**
CPU_CM.002	**Watchpoint PC functions can report false execution**	X		**10**
CPU_CM.003	**Prefetch faulting instructions can erroneously trigger breakpoints**	X		**11**
Firmware_CM.002	**Calculate Target Level for Temperature Comparison User Routine returns zero for valid temperature input parameter**	X		**12**
NVM_CM.001	**NVM Write access to trigger NVM erase operation must NOT be executed from NVM**	X		**12**

Table 4 **Functional Deviations** (cont'd)

Functional Deviation	Short Description	XMC1100	Chg	Pg
NVM_CM.002	Completion of NVM verify-only operations do not trigger NVM interrupt	X		13
SCU_CM.019	Temperature Sensor User Routines in ROM	X		13
USIC_AI.014	No serial transfer possible while running capture mode timer	X		14
USIC_AI.017	Clock phase of data shift in SSC slave cannot be changed	X		14
USIC_AI.018	Clearing PSR.MSLS bit immediately deasserts the SELOx output signal	X		14

Table 5 **Application Hints**

Hint	Short Description	XMC1100	Chg	Pg
ADC_AI.H006	Ratio of Module Clock to Converter Clock	X		16
ADC_AI.H007	Ratio of Sample Time t_S to SHS Clock f_{SH}	X		16
ADC_AI.H009	ADC Operation with internal reference, lower supply voltage range	X		18
Firmware_CM.H002	Ensuring correct selection of RxD Pin in ASC Bootstrap Loader	X		18
SCU_CM.H001	Temperature Sensor Functionality	X		18
USIC_AI.H004	I2C slave transmitter recovery from deadlock situation	X		18

Table 6 Documentation Updates

Hint	Short Description	XMC1200	XMC1201	XMC1202	Chg	Pg
SCU_CM.D001	SCU_CM.D001	X	X	X	New	20

2 Functional Deviations

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

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, 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

None.

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.

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.

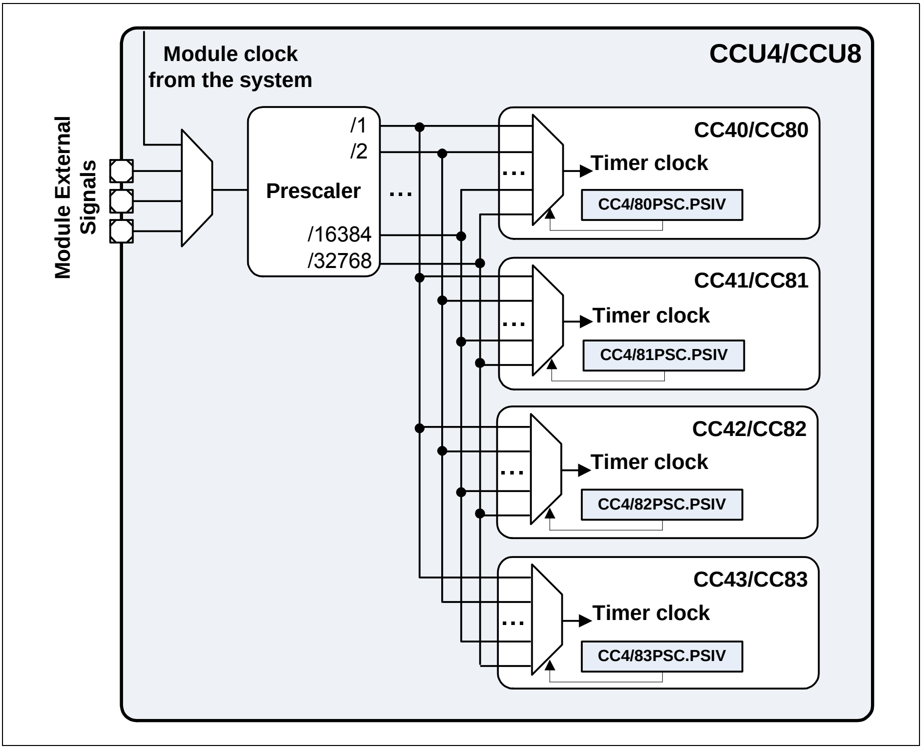

Figure 1 Clock Selection Diagram for CCU4/CCU8

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

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.

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.

SCU_CM.019 Temperature Sensor User Routines in ROM

These temperature sensor user routines in ROM cannot be used for EES and partial ES. For ES, the affected devices are identifiable through a 2-byte User Configuration Sector version 0002_H, stored in Flash Configuration Sector 0 (CS0), address $10000FEA_H$.

- Calculate Chip Temperature

- Calculate Target Level for Temperature Comparison

Workaround

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

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.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 7**.

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 7 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 x 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 $t_{ADCI} = t_{ADC} \times (DIVA+1)$

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.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 $480340BC_H$.

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.

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.

4 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.

SCU_CM.D001 DCO nominal frequencies and accuracy based on Temperature Sensor calibration

These parameters of the 64 MHz DCO1 Characteristics and 32 kHz DCO2 Characteristics tables in the XMC1000 family Data Sheet V1.4 based on AA-step are not valid for the AB-step Data Sheet.

- The accuracy of DCO1 based on temperature sensor calibration parameter, Δf_{LTT}.
- The min and max limits for DCO1 and DCO2 nominal frequency, f_{NOM} under nominal conditions after trimming. These limits are defined by the specified accuracy parameter over temperature Δf_{LT}.

Documentation Update

These parameters are not presented in the XMC1000 family AB-step Data Sheet V1.6. To improve the accuracy of the DCO1 oscillator, refer to XMC1000 Oscillator Handling Application Note.