Errata Sheet

Rel. 1.2, 2013-08

Device	**XMC1100**
Marking/Step	**EES-AA, ES-AA**
Package	**PG-TSSOP-16/38**

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	V1.0	Mar 2013
XMC1100 Data Sheet	V1.0	Aug 2013

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 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	2013-02	Initial Version
1.1	2013-03	Updated : ADC_AI.003
1.2	2013-03	Changes as described in **Table 4** and**Table 5**

Table 3 Errata fixed in this step

Errata	Short Description	Change
- none -		

Table 4 Functional Deviations

Functional Deviation	Short Description	XMC1100	Chg	Pg
ADC_AI.003	Additonal bit to enable ADC function	X		6
ADC_AI.004	ADC startup calibration sequence	X	Updated	6
ADC_AI.008	Wait-for-Read condition for register GLOBRES not detected in continuous auto-scan sequence	X	New	7
ADC_AI.010	ADC Operating Range	X	New	7
ADC_AI.011	Hardware Calibration Not Functional	X	New	7
CCU_AI.005	CCU4 and CCU8 External IP clock Usage	X		8
CPU_CM.002	Watchpoint PC functions can report false execution	X		10
CPU_CM.003	Prefetch faulting instructions can erroneously trigger breakpoints	X		11

Table 4 Functional Deviations (cont'd)

Functional Deviation	Short Description	XMC1100	Chg	Pg
Firmware_CM.001	User routine _NvmProgVerify stalls the system bus for two to three maximum 10 µs periods	X		12
PORTS_CM.004	Outputs of CCU4, BCCU and ACMP cannot be used to effectively control the pull devices on Pin	X		12
SCU_CM.010	Handling of Master Reset via bit RSTCON.MRSTEN	X		13
SCU_CM.011	Incomplete Initialisation after a System Reset	X		13
SCU_CM.012	Calibrating DCO function is not available	X	Updated	13
SCU_CM.013	Brownout reset triggered by External Brownout Detector (BDE)	X		14
SCU_CM.014	Temperature Sensor User Routines in ROM	X	New	14
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		15
USIC_AI.018	Clearing PSR.MSLS bit immediately deasserts the SELOx output signal	X		15

Table 5 Application Hints

Hint	Short Description	XMC1100	Chg	Pg
ADC_AI.H006	Ratio of Module Clock to Converter Clock	X	New	16
ADC_AI.H007	Ratio of Sample Time t_S to SHS Clock f_{SH}	X	New	16
NVM_CM.H001	Adding a wait loop to stand-alone verification sequences	X		18

2 Functional Deviations

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

ADC_AI.003 Additonal bit to enable ADC function

The analog section of ADC is not fully functioning when it is enabled by bit SHSCFG.ANOFF.

Workaround

To enable the analog section of the ADC, write value 00000001_H to register address 40010500_H in addition to the setup as mentioned above.

ADC_AI.004 ADC startup calibration sequence

In the AA step, the ADC startup calibration via GLOBCFG.SUCAL is incomplete.

Workaround

The automatic post calibration function need to be disabled during calibration and a software triggered re-calibration sequence is proposed.

The proposed startup calibration sequence is :

- Set bit GLOBCFG.SUCAL, GLOBCFG.DPCAL0 and GLOBCFG.DPCAL1 to 1
- Wait for the startup calibration to complete by polling bit SHSCFG.STATE
- Write value 80008000_H to register address $480340E0_H$ and $480340E4_H$

The proposed software triggered re-calibration sequence is :

- Write value 00000000_H to register address $480340E0_H$ and $480340E4_H$
- Set bit GLOBCFG.SUCAL, GLOBCFG.DPCAL0 and GLOBCFG.DPCAL1 to 1
- Wait for the calibration to complete by polling bit SHSCFG.STATE
- Write value 80008000_H to register address $480340E0_H$ and $480340E4_H$

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.010 ADC Operating Range

In the AA step, ADC operation below 4.5V is not supported. It is recommended to use the ADC within the 4.5V to 5.5V range.

Workaround

None.

ADC_AI.011 Hardware Calibration Not Functional

The built-in hardware calibration may not work under certain circumstances.

In this case, the result values will be inaccurate.

Workaround

After the startup calibration, disable postcalibration for all groups and replace this by the following software calibration routine:

- Clear the offset calibration value
- Convert channel CH31 (this is VAREF)
- If the result is below 0xFFF, decrease the gain calibration value:
  LOCi = LOCi + 0x00008000 - 1,
  otherwise increase it:
  LOCi = LOCi + 0x00008000 + 1.

The memory address of LOCi (i = 0 - 1) are:

480340C0$_H$ for group 0 (i = 0),

480340C4$_H$ for group 1 (i = 1).

<u>CCU_AI.005</u> 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.

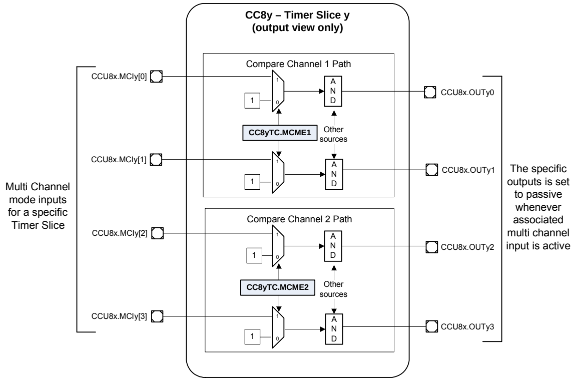

Figure 1 Clock Selection Diagram for CCU4/CCU8

Workaround

None.

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.

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

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 function is not available

The function of calibrating DCO based on temperature will not be supported in EES and ES sample.

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. Therefore, the temperature sensor function cannot be supported in the EES and ES samples.

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.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 6**.

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 6 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) x 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.

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.