Errata Sheet

Rel. 1.1, 17.01.2013

Device	**XC878LM Series**
Marking/Step	**AC**
Package	**PG-LQFP-64**

This Errata Sheet describes the deviations from the current user documentation.The module oriented classification and numbering system uses an ascending sequence over several derivatives, including already solved deviations. So gaps inside this enumeration can occur.

This Errata Sheet covers the following devices:

- XC878-13/16FF
- XC878M-13/16FF
- XC878LM-13/16FF

Table 1 Current Documentation

XC878CLM User's Manual	V1.1	Apr 2009
XC87xCLM Data Sheet	V1.5	Mar 2011

Each erratum identifier follows the pattern **Module_Arch.TypeNumber**:

- **Module**: subsystem or peripheral affected by the erratum
- **Arch**: microcontroller architecture where the erratum was firstly detected.
  - **AI**: Architecture Independent (detected on module level)
  - **CIC**: Companion ICs
  - **TC**: TriCore (32 bit)
  - **X**: XC1xx / XC2000 (16 bit)
  - **XC8**: XC800 (8 bit)
  - **none**: C16x (16 bit)

- **Type**: none - Functional Deviation; '**P**' - Parametric Deviation; '**H**' - Application Hint; '**D**' - Documentation Update
- **Number**: ascending sequencial number within the three previous fields. As this sequence is used over several derivatives, including already solved deviations, gaps inside this enumeration can occur.

Note: Devices marked with EES or ES are engineering samples which may not be completely tested in all functional and electrical characteristics, therefore they should be used for evaluation only.

The specific test conditions for EES and ES are documented in a separate Status Sheet.

1 History List / Change Summary

Table 2 History List

Version	Date	Remark
1.0	08.06.2010	

Table 3 Errata fixed in this step

Errata	Short Description	Chg
T2CCU_XC8.001	External Trigger of ADC when CCT of T2CCU overflows	Fixed

Table 4 Functional Deviations

Functional Deviation	Short Description	Chg	Pg
BROM_XC8.006	IRAM data is corrupted after any warm reset		6
CD_XC8.001	Set and Clear of Error Bit in CORDIC Linear Vectoring Mode		6
CD_XC8.002	Data Fetch to CD_STATC Register may capture an incorrect error status		7
SYS_XC8.001	MOV (direct, direct) instruction might cause a wrong value to be written to the destination register		7
SYS_XC8.003	Brownout Reset	New	13
T2CCU_XC8.003	T2CCU Capture Functions		13

Table 4 Functional Deviations (cont'd)

Functional Deviation	Short Description	Chg	Pg
UART_XC8.002	Bits FDEN and FDM in UART1_FDCON SFR cannot be Written by Read-Modify-Write Instructions		13

Table 5 Deviations from Electrical- and Timing Specification

AC/DC/ADC Deviation	Short Description	Chg	Pg

Table 6 Application Hints

Hint	Short Description	Chg	Pg
ADC_XC8.H001	Arbitration mode when using external trigger at the selected input line REQTR		16
BROM_XC8.H001	SYSCON0.RMAP handling in ISR		16
CCU6_XC8.H001	Pin Configuration of CCU6 Functions		17
CCU6_XC8.H002	CCU6 PM event in center-aligned mode		17
EVR_XC8.H002	Enhancement for Noise Immunity		18
FLASH_XC8.H004	Disable TRAP When Calling ROM Flash Routine		19
LIN_XC8.H001	LIN BRK field detection logic		19
PIN_XC8.H001	Current over GPIO pin must not source V_{DDP} higher than 0.3V		20
PM_XC8.H001	Clock source selection before entering power-down mode		20

Table 6 Application Hints (cont'd)

Hint	Short Description	Chg	Pg
PM_XC8.H002	SAK product variant does not support power-down mode		21
SYS_XC8.H001	Usage of the Bit Protection Scheme		21
SYS_XC8.H002	External Clock switching routine after a WDT reset		21
SYS_XC8.H003	Effective write for Read-Modify-Write instructions of two bytes, one machine cycle	New	21
T2_XC8.H003	Accessing Timer 21 registers		22

2 Functional Deviations

BROM_XC8.006 IRAM data is corrupted after any warm reset

After any warm reset (i.e. reset without powering off the device), boot up via User Mode affects certain IRAM data.

The affected IRAM address ranges are:

(1) 00_H - 07_H

Workaround

None

CD_XC8.001 Set and Clear of Error Bit in CORDIC Linear Vectoring Mode

In linear vectoring mode, the Error bit of register CD_STATC is set immediately on detecting overflow. When detected between iterations – the Error status is not held internally till the end of the calculation.

As the Error bit is defined such that it is cleared on any read access to the register (e.g. JB BSY), SW checking of the Error bit only at the end of calculation may miss to detect an overflow error condition.

Workaround

Especially in linear vectoring mode, if the error condition setting of Error bit must be detected, any read access should be done on the whole CD_STATC register (e.g. MOV) and the Error bit checked in all read instances.

CD_XC8.002 Data Fetch to CD_STATC Register may capture an incorrect error status

The error bit CD_STATC.ERROR is defined such that the bit is cleared on any read access to the register. Therefore, it is necessary to perform a data fetch on the register and check for the error bit in order not to lose the error status.

However, if CORDIC is clocked at two times PCLK and the execution time of the read instruction is more than one machine cycle, multiple read accesses will be performed on the CD_STATC register and the error bit will be cleared by the time the CPU performs the final read. As a result, the CPU does not capture the correct error status.

There is no problem if the CORDIC is clocked at the same frequency as PCLK.

Workaround

The following workarounds can be used to avoid incorrect data fetching from the CD_STATC register:

- The PUSH dir and POP dir instructions can be used to read the CD_STATC register in all conditions;
- The following one-machine cycle MOV instruction can be used to read the CD_STATC register when the CORDIC is clocked at two times of PCLK:
  – MOV A, dir

SYS_XC8.001 MOV (direct, direct) instruction might cause a wrong value to be written to the destination register

The MOV (direct, direct) instruction (hex code 85_H) that access registers (direct address ranging from 80_H to FF_H), does not write the correct value of the source register to the destination register if the destination register is a register listed in the table below.

The source register can be any register from the direct address range 80_H to FF_H.

Table 7

Module	Register	SFR Address	RMAP	Page	Products Affected
SCU	IRCON0	$B4_H$	0	0	XC88x, XC878
	IRCON1	$B5_H$	0	0	XC88x, XC878
	IRCON2	$B6_H$	0	0	XC88x, XC878
	IRCON3	$B4_H$	0	3	All
	IRCON4	$B5_H$	0	3	All
	NMISR	BC_H	0	0	XC88x, XC878
	FDCON	$E9_H$	0	0	XC88x, XC878
	PMCON0	$B4_H$	0	1	XC88x, XC878
	OSC_CON	$B6_H$	0	1	XC88x, XC878
	PLL_CON	$B7_H$	0	1	XC88x
	MISC_CON	$E9_H$	0	1	XC88x, XC878
WDT	WDTCON	BB_H	1	-	XC88x, XC878
CORDIC	CD_STATC	$A0_H$	1	-	XC88x, XC878
MDU	MDUSTAT	$B0_H$	1	-	XC88x, XC878
SSC	CONH (Operating Mode)	AB_H	0	-	All
UART1	SCON	$C8_H$	1	-	XC88x, XC878
	FDCON	CC_H	1	-	XC88x, XC878
T2	T2CON	$C0_H$	0	-	All
T21	T2CON	$C0_H$	1	-	XC88x, XC878
OCDS	MMCR2	$E9_H$	1	-	All
	MMCR	$F1_H$	1	-	All
	MMSR	$F2_H$	1	-	All
	MMICR	$F4_H$	1	-	All

Table 7

Module	Register	SFR Address	RMAP	Page	Products Affected
T2CCU	CCTCON	$C6_H$	0	1	XC878
	COSHDW	$C0_H$	0	2	XC878
	COCON	$C0_H$	0	3	XC878
	CC0L	$C1_H$	0	2	XC878
	CC0H	$C2_H$	0	2	XC878
	CC1L	$C3_H$	0	2	XC878
	CC1H	$C4_H$	0	2	XC878
	CC2L	$C5_H$	0	2	XC878
	CC2H	$C6_H$	0	2	XC878
	CC3L	$C1_H$	0	3	XC878
	CC3H	$C2_H$	0	3	XC878
	CC4L	$C3_H$	0	3	XC878
	CC4H	$C4_H$	0	3	XC878
	CC5L	$C5_H$	0	3	XC878
	CC5H	$C6_H$	0	3	XC878
CCU6	CC63SRL	$9A_H$	0	0	All
	CC63SRH	$9B_H$	0	0	All
	MCMOUTSL	$9E_H$	0	0	All
	MCMOUTSH	$9F_H$	0	0	All
	CC60SRL	FA_H	0	0	All
	CC60SRH	FB_H	0	0	All
	CC61SRL	FC_H	0	0	All

Table 7

Module	Register	SFR Address	RMAP	Page	Products Affected
CCU6 (cont'd)	CC61SRH	FD_H	0	0	All
	CC62SRL	FE_H	0	0	All
	CC62SRH	FF_H	0	0	All
	T12PRL	$9C_H$	0	1	All
	T12PRH	$9D_H$	0	1	All
	T13PRL	$9E_H$	0	1	All
	T13PRH	$9F_H$	0	1	All
	T12DTCL	$A4_H$	0	1	All
	T12DTCH	$A5_H$	0	1	All
	TCTR0L	$A6_H$	0	1	All
	TCTR0H	$A7_H$	0	1	All
	T12MSELL	$9A_H$	0	2	All
	T12MSELH	$9B_H$	0	2	All
	IENL	$9C_H$	0	2	All
	IENH	$9D_H$	0	2	All
	INPL	$9E_H$	0	2	All
	INPH	$9F_H$	0	2	All
	PSLR	$A6_H$	0	2	All
	MCMCTR	$A7_H$	0	2	All
	TCTR2L	FA_H	0	2	All
	TCTR2H	FB_H	0	2	All
	MODCTRL	FC_H	0	2	All
	MODCTRH	FD_H	0	2	All
	TRPCTRL	FE_H	0	2	All
	TRPCTRH	FF_H	0	2	All
	PISEL0L	$9E_H$	0	3	All
	PISEL0H	$9F_H$	0	3	All
	PISEL2	$A4_H$	0	3	All

Table 7

Module	Register	SFR Address	RMAP	Page	Products Affected
CCU6 (cont'd)	T13L	FC_H	0	3	All
	T13H	FD_H	0	3	All
	CMPSTATH	FF_H	0	3	All
ADC	GLOBCTR	CA_H	0	0	All
	PRAR	CC_H	0	0	All
	LCBR	CD_H	0	0	All
	INPCR0	CE_H	0	0	All
	ETRCR	CF_H	0	0	All
	CHCTR0	CA_H	0	1	All
	CHCTR1	CB_H	0	1	All
	CHCTR2	CC_H	0	1	All
	CHCTR3	CD_H	0	1	All
	CHCTR4	CE_H	0	1	All
	CHCTR5	CF_H	0	1	All
	CHCTR6	$D2_H$	0	1	All
	CHCTR7	$D3_H$	0	1	All
	RCR0	CA_H	0	4	All
	RCR1	CB_H	0	4	All
	RCR2	CC_H	0	4	All
	RCR3	CD_H	0	4	All
	CHINPR	CD_H	0	5	All
	EVINPR	$D3_H$	0	5	All
	CRCR1	CA_H	0	6	All
	CRPR1	CB_H	0	6	All
	CRMR1	CC_H	0	6	All
	QMR0	CD_H	0	6	All

For example, in the sample code below, there are two MOV (direct, direct) instructions that write the value of one register into another. All the source and destination registers in these two instructions are from the direct address range 80_H to FF_H.

The P1_DATA register is not one of the affected registers listed in the table above and therefore, it is written with the correct value of the CC60SRL register. On the other hand, the CC60SRH register is one of the affected registers and therefore, it is written with the wrong value of the B register.

Sample Code:

```
interrupt:
MUL A, B
MOV CC60SRL, A
MOV P1_DATA, CC60SRL
MOV CC60SRH, B
RETI
```

Workaround

Instead of using the MOV (direct, direct) instruction, use other instructions or an intermediate variable to write to the targeted register.

For example, the two MOV (direct, direct) instructions in the earlier sample code can be replaced with MOV (direct, A) instructions (hex code $F5_H$). Both the P1_DATA and CC60SRH registers will now be written with the correct source register values.

Sample Code:

```
interrupt:
MUL A, B
MOV CC60SRL, A
MOV P1_DATA, A
XCH A, B
MOV CC60SRH, A
RETI
```

SYS_XC8.003 Brownout Reset

Brownout reset may not be triggered when the core supply voltage (VDDC) drops below operating limit. It is recommended to use an external voltage detector and perform a power-on reset when the core supply voltage drops below 2.2V (minimum).

Workaround

None.

T2CCU_XC8.003 T2CCU Capture Functions

Capture mode 1 is the only T2CCU capture mode in XC878 AC step. Capture mode 0 of T2CCU is not functioning. In mode 1, a capture will occur upon writing to the low byte of the corresponding channel capture register, CCxL.

Workaround

None.

UART_XC8.002 Bits FDEN and FDM in UART1_FDCON SFR cannot be Written by Read-Modify-Write Instructions

The bits FDEN and FDM in UART1_FDCON SFR are not updated when written with the read-modify-write instructions listed in the table below:

Table 8

Affected Read-Modify-Write Instructions	Hex Code
INC dir	05
DEC dir	15
ANL dir,A	52
ANL dir,#data	53
ORL dir,A	42
ORL dir,#data	43

Table 8

Affected Read-Modify-Write Instructions	Hex Code
XRL dir,A	62
XRL dir,#data	63
XCH A,dir	C5
DJNZ dir,rel	D5

Workaround

Use MOV instructions, except MOV dir, dir (Hex Code: 85), when writing to the bits `FDEN` and `FDM` in `UART1_FDCON` SFR.

3 Deviations from Electrical- and Timing Specification

4 Application Hints

ADC_XC8.H001 Arbitration mode when using external trigger at the selected input line REQTR

If an external trigger is expected at the selected input line REQTR to trigger a pending request, the arbitration mode should be set (PRAR.ARBM=1) where the arbitration is started by pending conversion request. This selection will minimize the jitter between asynchronous external trigger with respect to the arbiter and the start of the conversion. The jitter can only be minimized while no other conversion is running and no higher priority conversion can cancel the triggered conversion. In this case, a constant delay (no jitter) has to be taken into account between the trigger event and the start of the conversion.

BROM_XC8.H001 SYSCON0.RMAP handling in ISR

The ISR has to handle SYSCON0.RMAP correctly when Flash user routines provided in the Boot ROM are used together with the interrupt system. Any ISR with the possibility of interrupting these user routines has to do the following in the interrupt routine:

save the value of the RMAP bit at the beginning

restore the value before the exit

This is to prevent access of the wrong address map upon return to the Flash user routine since the RMAP bit may be changed within the interrupt routine. The critical point is when Flash user routines sets RMAP to ‘1’ and the interrupt occurs that needs RMAP at ‘0’ in the ISR.

Please note that NMI is an interrupt as well.

CCU6_XC8.H001 Pin Configuration of CCU6 Functions

Table 9 shows the updated pin configuration of CCU6 pins in XC878.

Table 9 Updated CCU6 Pin Configuration

CCU6 Functions		Pin Symbol
CCPOS0_0	CCU6 Hall Input 0	P5.4
CCPOS1_0	CCU6 Hall Input 1	P5.3
CC60_3	Input of Capture/Compare Channel 0	P5.3
CC61_3	Input of Capture/Compare Channel 1	P5.4
CC62_3	Input of Capture/Compare Channel 2	P5.5

CCU6_XC8.H002 CCU6 PM event in center-aligned mode

After detecting a period match (PM A) in centre-aligned mode, T12 counts down from PM + 1 as shown below:

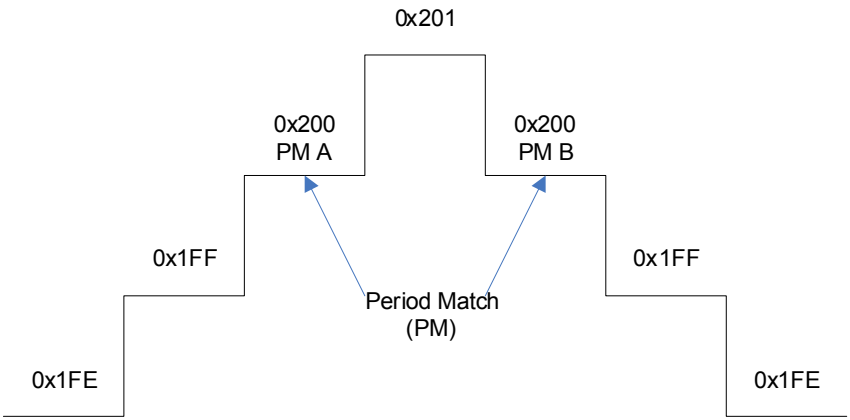

Figure 1 Counting sequence of T12 in center-aligned mode

This means a second PM event (PM B) will occur during the counting down. If ADC is triggered externally via ETRx2 (T12PM), it will be triggered twice in succession. Depending on how real-time the application code is running as well

as the T12 count rate and ADC conversion rate, the application could observe two ADC interrupts - once at PM A and once at PM B.

To avoid triggering twice the ADC interrupts, it is suggested to use ETRx6 from multi-channel mode instead of ETRx2 as the trigger source for ADC. Additional initialization are as follows:

- Configure MCMCTR.SWSEL = 101_B (Transfer on T12 period match)
- Configure MCMCTR.SWSYN = 00_B (Direct transfer)
- Write to MCMOUTSTL = CF_H (To enable multi-Channel PWM pattern on CC6x and COUT6x)

Note: Independent of the external trigger, the CCU6 internal triggers based on T12 PM (e.g. T12 PM interrupt or shadow transfer) are only activiated once while T12 is counting up.

EVR_XC8.H002 Enhancement for Noise Immunity

During power up, the EVR functionality may be affected due to injected noise from any functional pin. In order to enhance the noise immunity, the external reset pin RESET must be asserted until VDDC reaches 0.9 * VDDC. The delay of external reset can be realized by an external capacitor at RESET pin. This capacitor value must be selected so that VRESET reaches 0.4 V, but not before VDDC reaches 0.9 * VDDC.

A typical application example is shown in **Figure 2**. A 220 nF capacitor is connected to VDDP pin, VDDC pin and RESET pin. In addition, it is also essential to put it as close as possible to the chip.

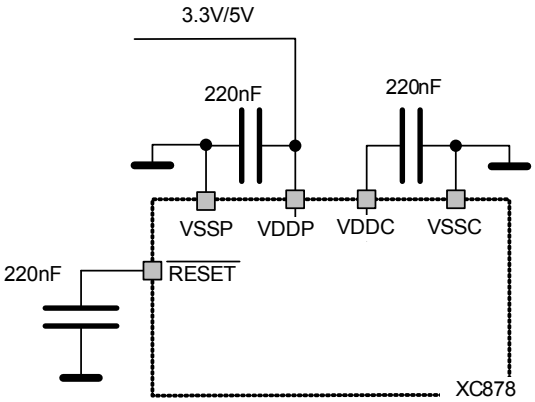

Figure 2 Noise Immunity Enhancement Circuitry

FLASH_XC8.H004 Disable TRAP When Calling ROM Flash Routine

User code has to disable TRAP by clearing EO.TRAP_EN bit before any CALL to ROM Flash routines, and to restore the bit after that.

LIN_XC8.H001 LIN BRK field detection logic

Based on the hardware implementation, the maximum number of bits in the BRK field must follow the formula:

$$\text{Maximum number of bits in BRK field} = \text{Baud Rate} \times \frac{4095}{\text{Sample Frequency}}$$

$$\text{Sample Frequency} = \frac{\text{PCLK}}{8 \times 2^{\text{BGSEL}}}$$

For example, if LIN baudrate is 19.2kbps, BGSEL = 0 and CPU frequency is 24MHz, the maximum number of bits in BRK field would be:

19.2k x 4095 / (24M / 8) = ~26.2 bits

If the maximum number of bits in the BRK field exceeded, the internal counter will overflow which results in baudrate detection error. Therefore, the user is advised to choose the appropriate BGSEL value for the required baudrate detection range.

The calculated value above does not consider sample error and transmission error, nevertheless it can be used as a guideline.

PIN_XC8.H001 Current over GPIO pin must not source V_{DDP} higher than 0.3V

When V_{DDP} is not powered on, the current over a GPIO pin has to be limited in such a way that V_{DDP} - V_{SSP} · 0.3V. This prevents the supply of the device via the ESD diode between the GPIO pin and V_{DDP}.

However, for applications with strict low power-down current requirements, it is mandatory that no active voltage source is supplied at any GPIO pin when V_{DDP} is not powered on.

PM_XC8.H001 Clock source selection before entering power-down mode

There are two oscillator sources available in the clock system: on-chip oscillator and external oscillator via XTAL pad. When external oscillator is selected to be the clock source (OSC_CON.OSCSS=1), the XTAL pad will not be shut down automatically during power-down mode. If optimal power-down current is required, on-chip oscillator should be chosen as the clock source before entering power-down mode.

Note: SAK product variant does not support power-down mode.

PM_XC8.H002 SAK product variant does not support power-down mode

Power-down mode is not available in the SAK product variant. It is only supported in SAF and SAX product variants. The profile of these variants is described in **Table 10**.

Table 10 Temperature Profile

Variant Type	Temperature Profile (°C)
SAF	-40 to 85
SAX	-40 to 105
SAK	-40 to 125

SYS_XC8.H001 Usage of the Bit Protection Scheme

When the bit protection scheme is enabled, bit field `PASSWD.PASS` should always be used to open and close write access to the protected bits. The scheme should be disabled only if it is not required in the application.

In the unlikely event that the scheme is enabled again after disabling it while the write access is still open, the write access will remain open until the count of 32 CCLK cycles is completed.

SYS_XC8.H002 External Clock switching routine after a WDT reset

When a WDT reset happens, the clock system will not be reset. If user needs to run the clock switching routine after a WDT reset, the bit field `OSC_CON.OSCSS` should be clear to 0 before activating the routine.

SYS_XC8.H003 Effective write for Read-Modify-Write instructions of two bytes, one machine cycle

When read-modify-write instructions requiring 2 bytes and 1 machine cycle (equivalent to 2 CCLK cycles) for execution, such as INC dir, are executed from memories without any wait states[1)], the actual write to the destination is delayed

by the internal bus for up to one CCLK cycle. This means that even though the CPU completes the instruction execution after 2 CCLK cycles, the write through the internal bus may take effect only after a further CCLK cycle.

The list of affected read-modify-write instructions is shown below:

Table 11

Mnemonic	Hex Code	Bytes	No. of CCLK cycles (without wait states)
INC dir	05	2	2
DEC dir	15	2	2
ANL dir, A	52	2	2
ORL dir, A	42	2	2
XRL dir, A	62	2	2
XCH A, dir	C5	2	2
CLR bit	C2	2	2
SETB bit	D2	2	2
CPL bit	B2	2	2

T2_XC8.H003 Accessing Timer 21 registers

To access Timer 21 registers, T2 page needs to be setup by clearing bit field `T2_PAGE.PAGE` to 000_B.

1) Applicable also to Flash memory with parallel read feature.