SIEMENS

Microcomputer Components
Technical Support Group Munich
HL DC AT

Errata Sheet

January 27, 1998 / Release 1.2

Device : **SAB-C167E2**

Stepping Code / Marking : **ES-AB**

The C167E2 is the bondout version for all C166 family microcontrollers (except 8xC166), supporting C167, C165, C164, C163 and C161 derivatives.

This errata sheet describes the functional problems known in this step. Problem classification and numbering is performed relative to modules, where the C167 AC-step is the reference. Since most problems of earlier steps have already been fixed in this step of the C167E2, problem numbering is not necessarily consecutive.

Changes from Errata Sheet **Rel. 1.0** for **C167E** devices with stepping code/marking **BA** to this Errata Sheet **Rel. 1.0** for **C167E2** devices with stepping code/marking **AA:**

- problems CPU.7, CPU.8, CPU.9, CPU.10, CPU.11, CPU.12 fixed
- problems ADC.7, X10, DC.4, DC.6, DC.7 fixed
- Power on Reset problem (RST.4)
- Disabling of CAN2 via XPERCON.1 (BX12)
- Access to external memory while emulated internal ROM/Flash/OTP is disabled (BROM.1)
- Clock Configuration during reset (BP0.1)
- CAPCOM6 Emergency Interrupt Vector Location (BINT.9)

Functional Problems

The following malfunctions are known in this step:

PWRDN.1: Execution of PWRDN Instruction while pin NMI# = high

When instruction PWRDN is executed while pin NMI# is at a high level, power down mode should not be entered, and the PWRDN instruction should be ignored. However, under the conditions described below, the PWRDN instruction may not be ignored, and no further instructions are fetched from external memory, i.e. the CPU is in a quasi-idle state. This problem will only occur in the following situations:

a) the instructions following the PWRDN instruction are located in external memory, and a **muliplexed bus** configuration **with memory tristate waitstate** (bit MTTCx = 0) is used, or

b) the instruction preceeding the PWRDN instruction **writes** to external memory or an XPeripheral (XRAM, CAN), and the instructions following the PWRDN instruction are located in external memory. In this case, the problem will occur for any bus configuration.

Note: the on-chip peripherals are still working correctly, in particular the Watchdog Timer will reset the device upon an overflow. Interrupts and PEC transfers, however, can not be processed. In case NMI# is asserted low while the device is in this quasi-idle state, power down mode is entered.

Workaround:

Ensure that no instruction which writes to external memory or an XPeripheral preceeds the PWRDN instruction, otherwise insert e.g. a NOP instruction in front of PWRDN. When a muliplexed bus with memory tristate waitstate is used, the PWRDN instruction should be executed out of internal RAM or XRAM.

CPU.17: Arithmetic Overflow by DIVLU instruction

For specific combinations of the values of the dividend (MDH,MDL) and divisor (Rn), the Overflow (V) flag in the PSW may not be set for **unsigned divide** operations, although an overflow occured.

E.g.:

MDH	MDL		Rn		MDH MDL
F0F0	0F0Fh	:	F0F0h	=	FFFF FFFFh, but no Overflow indicated !
					(result with 32-bit precision: 1 0000h)

The same malfunction appears for the following combinations:

n0n0 0n0n : n0n0
n00n 0nn0 : n00n
n000 000n : n000
n0nn 0nnn : n0nn where n means any Hex Digit between 8 ... F

i.e. all operand combinations where at least the most significat bit of the dividend (MDH) and the divisor (Rn) is set.

In the cases where an overflow occurred after DIVLU, but the V flag is not set, the result in MDL is equal to FFFFh.

Workaround:

Skip execution of DIVLU in case an overflow would occur, and explicitly set V = 1.

```
E.g.:           CMP Rn, MDH
                JMPR cc_ugt, NoOverflow ; no overflow if Rn > MDH
                BSET V                  ; set V = 1 if overflow would occur
                JMPR cc_uc, NoDivide    ; and skip DIVLU
NoOverflow:     DIVLU Rn
NoDivide: ...                   ; next instruction, may evaluate correct V flag
```

Note:

- the KEIL C compiler, run time libraries and operating system RTX166 do not generate or use instruction sequences where the V flag in the PSW is tested after a DIVLU instruction.

- with the TASKING C166 compiler, for the following intrinsic functions code is generated which uses the overflow flag for minimizing or maximizing the function result after a division with a DIVLU:

_div_u32u16_u16()
_div_s32u16_s16()
_div_s32u16_s32()

Consequently, an incorrect overflow flag (when clear instead of set) might affect the result of one of the above intrinsic functions but only in a situation where no correct result could be calculated anyway. These intrinsics first appeared in version 5.1r1 of the toolchain.

Libraries: not affected

CPU.16: Data read access with MOVB [Rn], mem instruction to internal program memory area

When the *MOVB [Rn], mem* instruction (opcode 0A4h) is executed, where

1. *mem* specifies a direct 16-bit byte operand address in the internal ROM/Flash/OTP memory area,

AND

2. *[Rn]* points to an **even** byte address, while the contents of the word which includes the byte addressed by *mem* is **odd,**

   OR

   [Rn] points to an **odd** byte address, while the contents of the word which includes the byte addressed by *mem* is **even**

the following problem occurs:

a) when *[Rn]* points to **external** memory or to the **X-Peripheral** (e.g. XRAM, CAN, etc.) address space, the data value which is written back is always 00h

b) when *[Rn]* points to the **internal** RAM or SFR/ESFR address space,

- the (correct) data value *[mem]* is written to *[Rn]***+1**, i.e. to the **odd** byte address of the selected word in case *[Rn]* points to an **even** byte address,
- the (correct) data value *[mem]* is written to *[Rn]***-1**, i.e. to the **even** byte address of the selected word in case *[Rn]* points to an **odd** byte address.

Workaround:

When *mem* is an address in internal ROM/Flash/OTP, substitute instruction

MOVB [Rn], mem e.g. by *MOV Rm, #mem*
MOVB[Rn], [Rm]

Note: the Keil C166 Compiler V3.10 has been extended by the directive FIXROM which avoids accesses to 'const' objectes via the instruction MOVB [Rn], mem.

RST.4 : Power-on Reset

Under certain circumstances some devices may show a problem during power-on concerning the internal clock and the RSTIN# signal.

When this problem occurs, the internal clock path between the XTAL1 input and the internal CPU clock is interrupted. In this case the whole internal circuitry gets no clock even if the oscillator is running or an external clock is available. The problem is caused by an undefinded input level (low or high) of an internal switch which connects the clock source to the internal CPU clock. The consequence is that the internal reset sequence and program execution cannot be started. In many cases also the RSTIN# signal is actively driven low until power is switched off.

The problem occurs sporadically during power-on. Not all devices show the problem because depending on the preference level of the internal switch, which depends on small technological variations of the production line, the clock source is connected to the internal CPU clock.

Workaround:

If the system does not start because the power-on reset problem has occured, then the supply voltage has to be switched off for a duration that Vcc reaches nearly 0 V at the microcontroller. Now supply voltage has to be switched on again. Up to now there is no hardware workaround known which can eliminate the problem in the system.

Bondout Specific Functional Problems:

BROM.1: Access to external memory while emulated internal ROM/Flash/OTP is disabled

When the emulated internal ROM/Flash/OTP is disabled (pin EA# = low during reset and/or bit SYSCON.ROMEN = 0),
and in register EMUCON a ROM size > 128 Kbyte is selected,
and an access to a location **s:Fx**XXh (**x** = 200h .. DFFh, **s**: see table below) is performed,
then the emulated internal ROM/Flash/OTP will be accessed instead of external memory under the following conditions:

Selected ROM size	Selected segment s
512 Kbyte	3, 4, 5, 6, 7, 8
384 Kbyte	3, 4, 5, 6
256 Kbyte	3, 4
196 Kbyte	3

Workaround:
Do not select a ROM size > 128 Kbyte in register EMUCON when the internal ROM/Flash/OTP shall not be emulated in an application.

BP0.1: Clock Configuration during Reset

When CLKCFG = 001 is selected for P0H[7:5] during reset, this will result in an internal CPU clock fcpu = fxtal/4 instead of fxtal/2.

BINT.9: Emergency Interrupt Vector Location

When a CAPCOM6 Emergency interrupt request is generated, it will branch to location 011Ch (same as ASC0 Transmit Buffer interrupt) instead of 013Ch.

Workaround:
When both the CAPCOM6 Emergency Interrupt and the ASC0 Transmit Buffer Interrupt are used, a CAPCOM6 Emergency Interrupt may be distinguished by software from an ASC0 Transmit Buffer Interrupt by the condition
(TRF = 1 or BCERR = 1) and CC6EIR = 0

Functional Problem	Short Description	Remarks
PWRDN.1	Execution of PWRDN Instruction while pin NMI# = high	
CPU.17	Arithmetic Overflow by DIVL Instructions	
CPU.16	Data read access with MOVB [Rn], mem instruction to internal ROM/Flash/OTP area	
RST.4	Power on Reset	
BROM.1	Access to external memory while emulated internal ROM/Flash/OTP is disabled	
BP0.1	Clock Configuration during reset	
BINT.9	CAPCOM6 Emergency Interrupt Vector Location	

Table 1: Functional Problems of the C167E2

Specific Problems with X-Peripherals (XPERs)

The following problems with the interface to XPERs, the CAN module, and the XRAM module are currently known:

X9: Read Access to XPERs in Visible Mode

The data of a read access to an XBUS-Peripheral (XRAM, CAN) in Visible Mode is not driven to the external bus. PORT0 is tristated during such read accesses.

BX12: Disabling of CAN2 via XPERCON.1

Disabling of CAN2 via XPERCON.1 does not work: CAN2 is always enabled when bit SYSCON.XPEN = 1.

Functional Problem	Short Description	fixed in Step
BX12	Disabling of CAN2 via XPERCON.1	
X9	Read Access to XPERs in Visible Mode	

Table 2: Functional Problems with XPERs on the C167E2

Deviations from DC/AC Specification

Notes:

1) Pin READY# has an internal pullup (all C167xx derivatives). This will be documented in the next revision of the Data Sheet.

2) Timing t28: Parameter description and test changed from ’Address hold after RD#/WR#’ to ’Address hold after WR#’. It is guaranteed by design that read data are internally latched by the controller before the address changes.

3) During reset, the internal pullups on P6.[4:0] are active, independent whether the respective pins are used for CS# function after reset or not.