April 21, 2009

C8051F930/31/20/21 Rev. B/C/D Errata

Errata Status Summary

Errata #	Title	Status		
		Rev B Device	Rev C Device	Rev D Device
1	Maximum clock speed	See Errata Details 1	Fixed	Fixed
2	Increased device leakage for supply voltage above 3.0 V	See Errata Details 2	Fixed	Fixed
3	SmaRTClock capture interferes with auto reset function	See Errata Details 3	Fixed	Fixed
4	SYNC bit does not always provide best SNR performance	See Errata Details 4	Fixed	Fixed
5	Entry into debug mode upon wakeup from sleep	See Errata Details 5	Screened During Production Test	Fixed
6	Increased supply current in one-cell mode for VBAT < 0.9 V and VDD/DC+ > 2.7 V	See Errata Details 6	See Errata Details 6	Fixed
7	Development tool update to support Run/Stop function in Sleep and Suspend Modes	Not Required	Not Required	Required See Errata Details 7
8	POR threshold calibration value	Not Applicable	See Errata Details 8	Fixed
9	SMBus Hardware ACK behavior	See Errata Details 9	See Errata Details 9	See Errata Details 9
10	Writes to CRC0CN that initiate a CRC0 operation	See Errata Details 10	See Errata Details 10	See Errata Details 10
11	Writes to FLSCL that enable the Flash read one-shot timer	See Errata Details 11	See Errata Details 11	See Errata Details 11
12	Reading PMU0CF upon wakeup from Suspend Mode	See Errata Details 12	See Errata Details 12	See Errata Details 12
13	System clock setting upon entry into Suspend or Sleep Mode	See Errata Details 13	See Errata Details 13	See Errata Details 13

Rev D is available for prototyping and production.

Errata Details

1. **Maximum clock speed**

   Description: A timing path which does not allow the device to operate at 25 MHz under all specified operating conditions has been identified.

   Impacts: The following restrictions regarding the system clock frequency should be followed:
   - If the 24.5 MHz Precision Oscillator is selected as the system clock source, the device may be operated in temperatures up to 50 °C. For operation at higher temperatures, the global system clock divider should be set to the divide by 2 setting.
   - If the External Oscillator is selected as the system clock source, the maximum system clock frequency is 22 MHz when the duty cycle is between 45 % and 55 %.
   - There are no restrictions when using the Low Power Oscillator or the smaRTClock oscillator as the system clock source.

   Resolution: Fixed in Rev C and later.

2. **Increased device leakage for supply voltage above 3.0 V**

   Description: An incorrectly wired device that causes increased leakage at supply voltages above 3.0 V has been identified.

   Impacts: The measured supply current in Sleep Mode increases at supply voltages above 3.0 V.

   Resolution: Fixed in Rev C and later.

3. **SmaRTClock capture interferes with auto reset function**

   Description: Capture of the smaRTClock timer interferes with the auto reset function.

   Impacts: The capture and auto reset functions of the smaRTClock timer cannot be used simultaneously.

   Resolution: Fixed in Rev C and later.

4. **SYNC bit does not always provide best SNR performance**

   Description: Under some operating conditions, the SYNC bit in the DC0CN register does not provide the best SNR performance for the 10-bit SAR ADC.

   Impacts: When the device is powered from the dc-dc converter, setting the SYNC bit will synchronize the clocking of the ADC to that of the dc-dc converter. On Rev B devices, only the ADC tracking period is synchronized with the dc-dc converter and under some operating conditions the ADC SNR can still be degraded by dc-dc switching activity.

   Resolution: Fixed in Rev C and later. For Rev C and later, setting the SYNC bit will synchronize both the tracking and conversion periods of the ADC to the dc-dc converter clock.

5. **Entry into debug mode upon wakeup from sleep**

   Description: A certain percentage of devices are exhibiting a behavior which causes the MCU to enter a debug state upon wakeup from sleep mode.

   Impacts: Devices exhibiting this behavior may stop executing code or may output internal digital signals onto GPIO pins upon wakeup from sleep mode.

   Resolution: Rev C production test has been updated to screen for devices having a high risk of exhibiting this behavior. Fixed in Rev D and later.

6. **Increased supply current in one-cell mode for VBAT < 0.9 V and VDD/DC+ > 2.7 V**

   Description: In one-cell mode, if the VDD/DC+ supply voltage (dc-dc converter output) is greater than 2.7 V and the VBAT supply voltage (dc-dc converter input) is less than 0.9 V, the device supply current may reach an increased level until power is cycled.

   Impacts: If powered by a battery, the battery may be discharged at a faster rate as soon as its voltage drops below 0.9 V.

   Workaround: Monitor the VBAT supply voltage and place the device in a low power state before the battery voltage drops below 0.9 V if the dc-dc converter is programmed to an output voltage greater than 2.7 V.

   Resolution: Fixed in Rev D and later.

7. **Development tool update to support Run/Stop function in Sleep and Suspend Modes**

   Description: On Rev D devices, a development tool update is required to support Run/Stop functionality while the MCU is in sleep or suspend mode.

   Impacts: Without the development tool update, the connection with the IDE/debugger will be disconnected if the user attempts to stop the MCU while it is in sleep or suspend mode.

   Resolution: When using Rev D and later production silicon the following development tool revisions are required to enable full debug capability:

   Silicon Labs IDE – Revision 3.41 or later

   Keil µVision Driver – Revision 2.81 or later

   Flash Programming Utilities – Revision 3.11 or later

   Production Programming Utility – Revision 1.41 or later.

8. **POR threshold calibration value**

   Description: On some Rev C devices, the POR threshold was calibrated to a value higher than the datasheet specification.

   Impacts: The device may enter the reset state at a voltage higher than the voltage specified in the datasheet.

   Resolution: Fixed in Rev D and later.

9. **SMBus Hardware ACK behavior**

Description: In some system management bus (SMBus) configurations, the Hardware Acknowledge mechanism of the SMBus peripheral can cause incorrect or undesired behavior. The Hardware Acknowledge mechanism is enabled when the EHACK bit (SMB0ADM.0) is set to logic 1.

The configurations to which this errata does not apply are:

a. All SMBus configurations when Hardware Acknowledge is disabled.
b. All **single-master / single-slave** SMBus configurations when Hardware Acknowledge is enabled and the MCU is operating as a **master or slave**.
c. All **multi-master / single-slave** SMBus configurations when Hardware Acknowledge is enabled and the MCU is operating as a **slave**.
d. All **single-master / multi-slave** SMBus configurations when Hardware Acknowledge is enabled and the MCU is operating as a **master**.

This errata only applies to the following configurations:

a. All **multi-slave** SMBus configurations when Hardware Acknowledge is enabled and the MCU is operating as a **slave**.
b. All **multi-master** SMBus configurations when Hardware Acknowledge is enabled and the MCU is operating as a **master**.

The following issues are present when operating as a **slave** in a **multi-slave** SMBus configuration:

a) When Hardware Acknowledge is enabled and SDA setup and hold times are not extended (EXTHOLD = 0 in the SMB0CF register), the SMBus hardware will always generate an SMBus interrupt following the ACK/NACK cycle of any slave address transmission on the bus, whether or not the address matches the conditions of SMB0ADR and SMB0MASK. The expected behavior is that an interrupt is only generated when the address matches.

b) When Hardware Acknowledge is enabled and SDA setup and hold times are extended (EXTHOLD = 1 in the SMB0CF register), the SMBus hardware will only generate an SMBus interrupt as expected when the slave address transmission on the bus matches the conditions of SMB0ADR and SMB0MASK. However, in this mode, the Start bit (STA) will be incorrectly cleared on reception of a slave address before software vectors to the interrupt service routine.

c) When Hardware Acknowledge is enabled and the ACK bit (SMB0CN.1) is set to 1, an unaddressed slave may cause interference on the SMBus by driving SDA low during an ACK cycle. The ACK bit of the unaddressed slave may be set to 1 if any device on the bus generates an ACK.

Impacts:

a) Once the CPU enters the interrupt service routine, SCL will be asserted low until SI is cleared, causing the clock to be stretched when the MCU is not being addressed. This may limit the maximum speed of the SMBus if the master supports SCL clock stretching. Incompliant SMBus masters that do not support SCL clock stretching will not recognize that the clock is being stretched. If the CPU issues a write to SMB0DAT, it will have no effect on the bus. No data collisions will occur.

b) Once the hardware has matched an address and entered the interrupt service routine, the firmware will not be able to use the Start bit to distinguish between the reception of an address byte versus the reception of a data byte. However, the hardware will still correctly acknowledge the address byte (SLA+R/W).

c) The SMBus master and the addressed slave are prevented from generating a NACK by the unaddressed slave because it is holding SDA low during the ACK cycle. There is a potential for the SMBus to lock up.

Workarounds:

a) The SMBus interrupt service routine should verify an address when it is received and clear SI as soon as possible if the address does not match to minimize clock stretching. To prevent clock stretching when not being addressed, enable setup and hold time extensions (EXTHOLD = 1).

b) Detection of Initial Start:

   To distinguish between the reception of an address byte **at the beginning** of a transfer versus the reception of a data byte when setup and hold time extensions are enabled (EXTHOLD = 1), software should maintain a status bit to determine whether it is currently inside or outside a transfer. Once hardware detects a matching slave address and interrupts the MCU, software should assume a start condition and set the software bit to indicate that it is currently inside a transfer. A transfer ends any time the STO bit is set or on an error condition (e.g. SCL Low Timeout).

   Detection of Repeated Start:

   To detect the reception of an address byte **in the middle** of a transfer when setup and hold time extensions are enabled (EXTHOLD = 1), disable setup and hold time extensions (EXTHOLD = 0) upon entry into a transfer and re-enable setup and hold time extensions (EXHOLD = 1) at the end of a transfer.

c) Schedule a timer interrupt to clear the ACK bit at an interval shorter than 7 bit periods when the slave is not being addressed. For example, on a 400 kHz SMBus, the ACK bit should be cleared every 17.5 uS (or at 1/7 the bus frequency, 57 kHz). As soon as a matching slave address is detected (a transfer is started), the timer which clears the ACK bit should be stopped and its interrupt flag cleared. The timer should be re-started once a stop or error condition is detected (the transfer has ended).

A code example demonstrating these workarounds can be found in the SMBus examples folder with the following default location:

C:\SiLabs\MCU\Examples\C8051F93x_92x\SMBus\F93x_SMBus_Slave_Multibyte_HWACK.c

The SMBus examples folder, along with examples for many additional peripherals, is created when the Silicon Laboratories IDE is installed. The latest version of the IDE may be downloaded from the software downloads page www.silabs.com/MCUDownloads on the Silicon Laboratories website.

The following issue is present when operating as a **master** in a **multi-master** SMBus configuration:

If the SMBus master loses arbitration in a multi-master system, it may cause interference on the SMBus by driving SDA low during the ACK cycle of transfers which it is not participating. This will occur regardless of the state of the ACK bit (SMB0CN.1) .

Impacts:

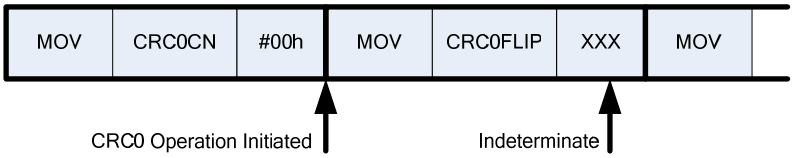
The SMBus master and slave participating in the transfer are prevented from generating a NACK by the MCU because it is holding SDA low during the ACK cycle. There is a potential for the SMBus to lock up.

Workaround:

Disable Hardware Acknowledge (EHACK = 0) when the MCU is operating as a master in a multi-master SMBus configuration.

10. **Writes to CRC0CN that initiate a CRC0 operation**

Description: The third op-code byte fetched from program memory following a write to CRC0CN that initiates a CRC0 operation is indeterminate.



Impacts: If the indeterminate op-code byte is the first or second byte in an instruction, improper code execution may result.

Workaround: Writes to CRC0CN that initiate a CRC0 operation must be immediately followed by a benign 3-byte instruction whose third byte is a don't care. An example of such an instruction is the write of a dummy value to the CRC0FLIP register using a 3-byte MOV instruction. The value written to CRC0FLIP will be indeterminate, but this should have no effect on the system. To ensure that both instructions are executed without interruption, global interrupts should be disabled.

Note: When programming in 'C', the dummy value written to CRC0FLIP should be a non-zero value. This prevents the compiler from generating the following instruction sequence:

```
CLR  A
MOV  CRC0FLIP,  A
```

When programming in 'C', the disassembly should be checked to ensure the compiler generated the following instruction sequence:

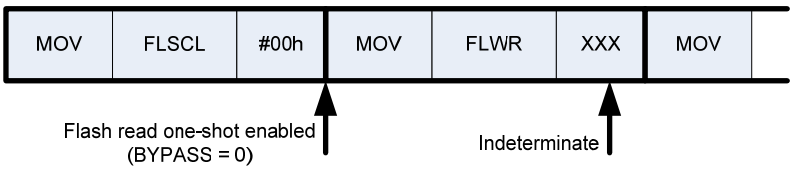
```
MOV  CRC0FLIP,  #AAh  where #AAh is the non-zero dummy value.
```

Resolution: This behavior will be described in the data sheet.

11. **Writes to FLSCL that enable the Flash read one-shot timer**

Description: The Flash read one-shot timer is enabled on reset and reduces supply current when operating at system clock frequencies below 10 MHz. When operating at system clock frequencies above 10 MHz, power consumption can be minimized if the one-shot timer is disabled by setting the BYPASS bit (FLSCL.6) to logic 1. The Flash read one-shot timer should be re-enabled by clearing the BYPASS bit (FLSCL.6) to logic 0 when entering a low power mode or if the system clock frequency is reduced below 10 MHz.

The third op-code byte fetched from program memory following a write to FLSCL that enables the Flash read one-shot timer is indeterminate.



Impacts: If the indeterminate op-code byte is the first or second byte in an instruction, improper code execution may result.

Workaround: Writes to FLSCL that enable the Flash read one-shot timer must be immediately followed by a benign 3-byte instruction whose third byte is a don't care. An example of such an instruction is the write of a dummy value to the FLWR register using a 3-byte MOV instruction. All writes to the FLWR register are don't care and have not effect on the system. To ensure that both instructions are executed without interruption, global interrupts should be disabled.

Note: When programming in 'C', the dummy value written to FLWR should be a non-zero value. This prevents the compiler from generating the following instruction sequence:

```
CLR  A
MOV  FLWR,  A
```

When programming in 'C', the disassembly should be checked to ensure the compiler generated the following instruction sequence:

```
MOV  FLWR,  #AAh   where #AAh is the non-zero dummy value.
```

Resolution: This behavior is described in the data sheet.

12. **Reading PMU0CF upon wakeup from Suspend Mode**

Description: Upon wakeup from Suspend Mode, the power management unit requires two system clocks in order to update the wake-up source flags in the PMU0CF register.

Impacts: If the PMU0CF register is read in the first two clock cycles following the wakeup from Suspend Mode, all wakeup source flags will contain a value of zero.

Workaround: When waking up from Suspend, wait at least 2 system clock cycles before reading the PMU0CF register to determine the cause of the wakeup.

Resolution: This behavior will be described in the data sheet.

13. **System clock setting upon entry into Suspend or Sleep Mode**

Description: Due to a clock synchronization issue inside the power management unit, if a wakeup event occurs during the same clock cycle that places the device into Sleep or Suspend Mode, then the device may partially enter the low power mode and remain in this state until the next power-on reset. The supply current in this state is approximately 2 mA.

Impacts: This behavior may cause the device to become unresponsive to all wake-up sources and to the reset pin. It can only be recovered with a power-on reset.

Workaround: Since the power management unit is clocked from the low power oscillator, setting the system clock to low power oscillator divided by 2 before entering Suspend or Sleep Mode will ensure that the clock synchronization issue inside the power management unit is not exercised and that the device will be able to safely enter and exit the low power state. The CLKRDY bit (CLKSEL.7) should be polled after changing the clock divide setting to ensure that the divide by 2 setting is applied to the system clock before allowing the device to enter the low power mode.