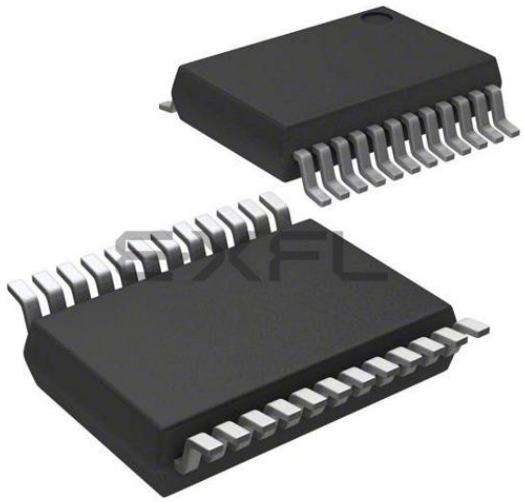

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Last Time Buy
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	0.9V ~ 3.6V
Data Converters	A/D 15x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	24-QSOP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f911-d-gu

List of Tables

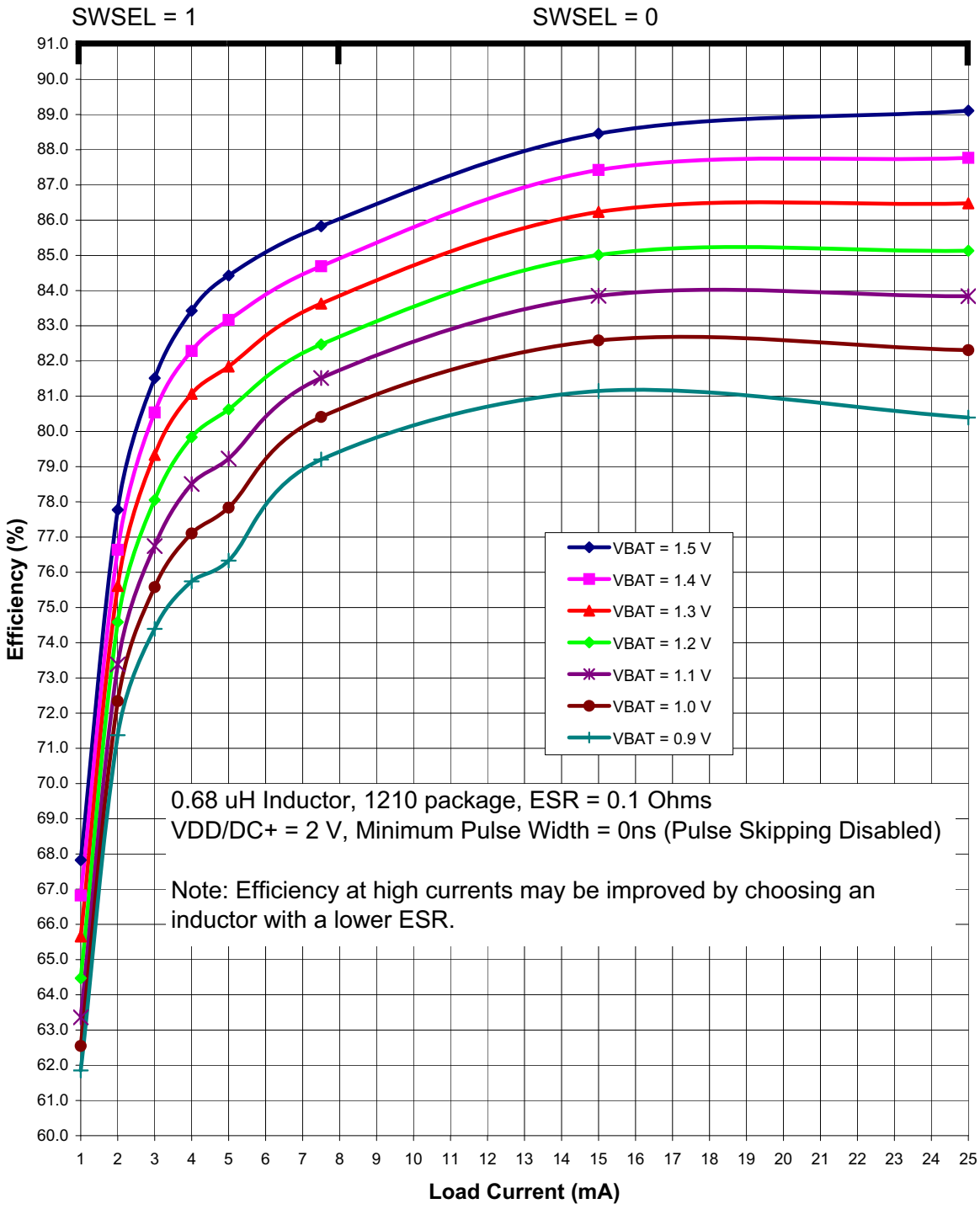

Figure 4.3. Typical DC-DC Converter Efficiency (High Current, VDD/DC+ = 2 V)

5.8.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 4.11 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.9 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

A single-point offset measurement of the temperature sensor is performed on each device during production test. The measurement is performed at 25 °C ±5 °C, using the ADC with the internal high speed reference buffer selected as the Voltage Reference. The direct ADC result of the measurement is stored in the SFR registers TOFFH and TOFFL, shown in SFR Definition 5.13 and SFR Definition 5.14.

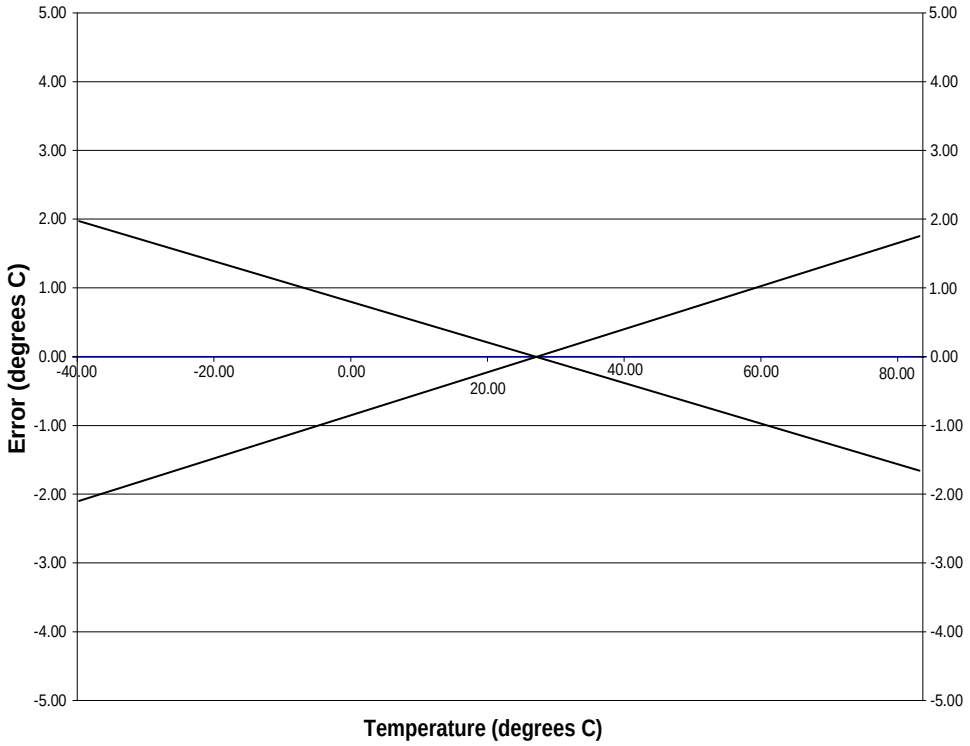

Figure 5.9. Temperature Sensor Error with 1-Point Calibration (V_{REF} = 1.68 V)

7.2. Comparator Outputs

When a comparator is enabled, its output is a logic 1 if the voltage at the positive input is higher than the voltage at the negative input. When disabled, the comparator output is a logic 0. The comparator output is synchronized with the system clock as shown in Figure 7.2. The synchronous “latched” output (CP0, CP1) can be polled in software (CPnOUT bit), used as an interrupt source, or routed to a Port pin (configured for digital I/O) through the Crossbar.

The asynchronous “raw” comparator output (CP0A, CP1A) is used by the low power mode wake-up logic and reset decision logic. See the Power Options chapter and the Reset Sources chapter for more details on how the asynchronous comparator outputs are used to make wake-up and reset decisions. The asynchronous comparator output can also be routed directly to a Port pin through the Crossbar, and is available for use outside the device even if the system clock is stopped.

When using a Comparator as an interrupt source, Comparator interrupts can be generated on rising-edge and/or falling-edge comparator output transitions. Two independent interrupt flags (CPnRIF and CPnFIF) allow software to determine which edge caused the Comparator interrupt. The comparator rising-edge and falling-edge interrupt flags are set by hardware when a corresponding edge is detected regardless of the interrupt enable state. Once set, these bits remain set until cleared by software.

The rising-edge and falling-edge interrupts can be individually enabled using the CPnRIE and CPnFIE interrupt enable bits in the CPTnMD register. In order for the CPnRIF and/or CPnFIF interrupt flags to generate an interrupt request to the CPU, the Comparator must be enabled as an interrupt source and global interrupts must be enabled. See the Interrupt Handler chapter for additional information.

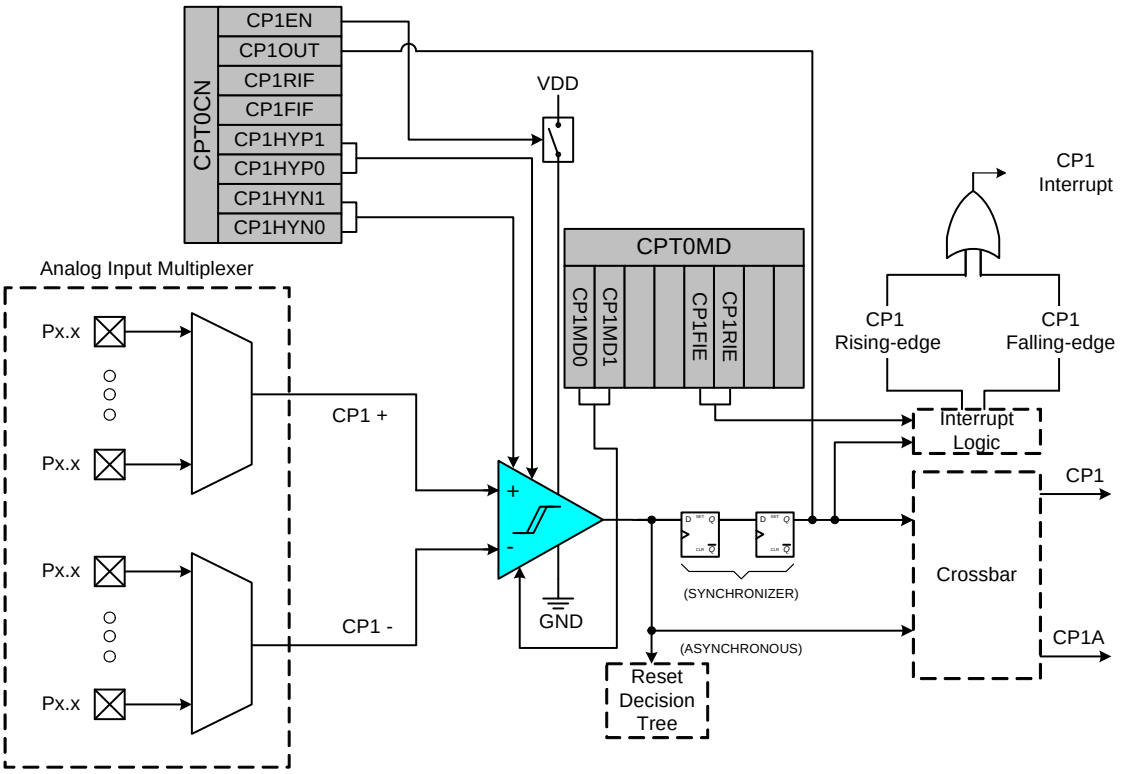

Figure 7.2. Comparator 1 Functional Block Diagram

8. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 27), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 8.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

8.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

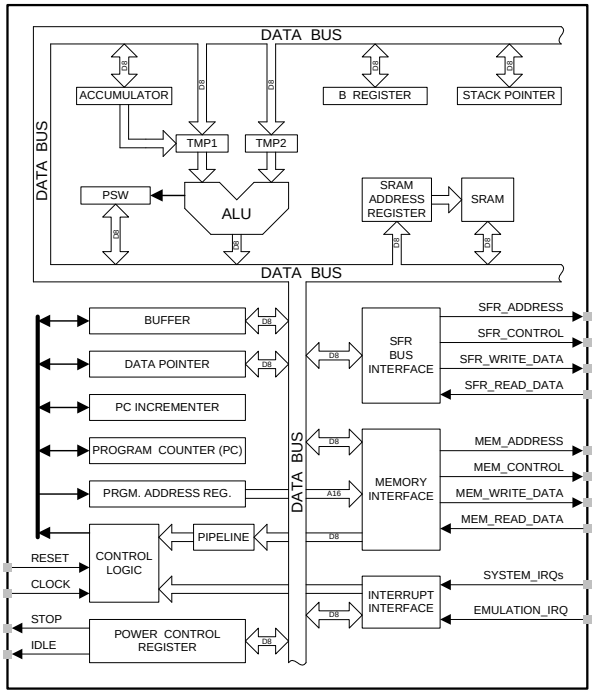

Figure 8.1. CIP-51 Block Diagram

12.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. If a high priority interrupt preempts a low priority interrupt, the low priority interrupt will finish execution after the high priority interrupt completes. Each interrupt has an associated interrupt priority bit in in the Interrupt Priority and Extended Interrupt Priority registers used to configure its priority level. Low priority is the default.

If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate. See Table 12.1 on page 122 to determine the fixed priority order used to arbitrate between simultaneously recognized interrupts.

12.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 7 system clock cycles: 1 clock cycle to detect the interrupt, 1 clock cycle to execute a single instruction, and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 19 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

SFR Definition 12.7. IT01CF: $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Page = 0x0; SFR Address = 0xE4

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7

Table 13.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	FEDR	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	FEDR	FEDR
Erase page containing Lock Byte - Unlock all pages (if any page is locked)	Only by C2DE	FEDR	FEDR
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	FEDR	FEDR
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	FEDR	FEDR
Read, Write or Erase Reserved Area	Not Permitted	FEDR	FEDR

C2DE—C2 Device Erase (Erases all Flash pages including the page containing the Lock Byte)
FEDR—Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is 1 after reset)

- All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).
- Locking any Flash page also locks the page containing the Lock Byte.
- Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.
- If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.
- The scratchpad is locked when all other Flash pages are locked.
- The scratchpad is erased when a Flash Device Erase command is performed.

13.5.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write Flash bytes and one routine in code that sets both PSWE and PSEE both to a 1 to erase Flash pages.
8. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in AN201, *"Writing to Flash from Firmware"*, available from the Silicon Laboratories web site.
9. Disable interrupts prior to setting PSWE to a 1 and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

13.5.3. System Clock

12. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.

16.1. Startup Behavior

On initial power-on, the dc-dc converter outputs a constant 50% duty cycle until there is sufficient voltage on the output capacitor to maintain regulation. The size of the output capacitor and the amount of load current present during startup will determine the length of time it takes to charge the output capacitor.

During initial power-on reset, the maximum peak inductor current threshold, which triggers the overcurrent protection circuit, is set to approximately 125 mA. This generates a “soft-start” to limit the output voltage slew rate and prevent excessive in-rush current at the output capacitor. In order to ensure reliable startup of the dc-dc converter, the following restrictions have been imposed:

- The maximum dc load current allowed during startup is given in Table 4.16 on page 60. If the dc-dc converter is powering external sensors or devices through the VDD/DC+ pin or through GPIO pins, then the current supplied to these sensors or devices is counted towards this limit. The in-rush current into capacitors does not count towards this limit.
- The maximum total output capacitance is given in Table 4.16 on page 60. This value includes the required 1 µF ceramic output capacitor and any additional capacitance connected to the VDD/DC+ pin.

Once initial power-on is complete, the peak inductor current limit can be increased by software as shown in Table 16.1. Limiting the peak inductor current can allow the device to start up near the battery’s end of life.

.

Table 16.1. IPeak Inductor Current Limit Settings

SWSEL	ILIMIT	Peak Current (mA) Normal Power Mode	Peak Current (mA) Low Power Mode
1	0	100	75
1	1	125	100
0	0	250	125
0	1	500	250

The peak inductor current is dependent on several factors including the dc load current and can be estimated using following equation:

$$I_{PK} = \sqrt{\frac{2\ I_{LOAD}(VDD/DC+ - VBAT)}{efficiency \times inductance \times frequency}}$$

efficiency = 0.80

inductance = 0.68 µH

frequency = 2.4 MHz

18.1. Power-On (VBAT Supply Monitor) Reset

During power-up, the device is held in a reset state and the $\overline{RST}$ pin is driven low until V_{BAT} settles above V_{POR}. An additional delay occurs before the device is released from reset; the delay decreases as the V_{BAT} ramp time increases (V_{BAT} ramp time is defined as how fast V_{BAT} ramps from 0 V to V_{POR}). Figure 18.3 plots the power-on and V_{DD} monitor reset timing. For valid ramp times (less than 3 ms), the power-on reset delay ($T_{PORDelay}$) is typically 3 ms (V_{BAT} = 0.9 V), 7 ms (V_{BAT} = 1.8 V), or 15 ms (V_{BAT} = 3.6 V).

Note: The maximum V_{DD} ramp time is 3 ms; slower ramp times may cause the device to be released from reset before V_{BAT} reaches the V_{POR} level.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000), software can read the PORSF flag to determine if a power-up was the cause of reset. The contents of internal data memory should be assumed to be undefined after a power-on reset.

On ‘F912 and ‘F902 devices, the VBAT supply monitor can be disabled to save power by writing ‘1’ to the MONDIS (PMU0MD.5) bit. When the VBAT supply monitor is disabled, all reset sources will trigger a full POR and will re-enable the VBAT supply monitor.

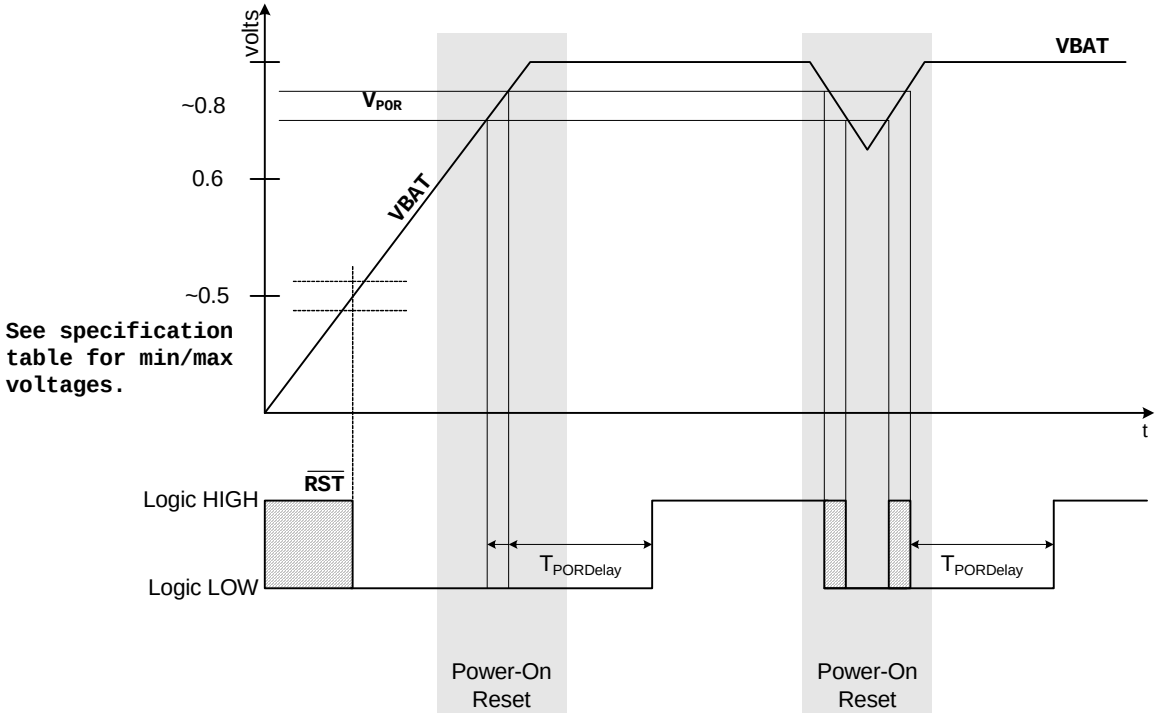

Figure 18.2. Power-Fail Reset Timing Diagram

SFR Definition 20.1. RTC0KEY: SmaRTClock Lock and Key

Bit	7	6	5	4	3	2	1	0
Name	RTC0ST[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAE

Bit	Name	Function
7:0	RTC0ST	**SmaRTClock Interface Lock/Key and Status.** Locks/unlocks the SmaRTClock interface when written. Provides lock status when read. **Read:** **0x00:** SmaRTClock Interface is locked. **0x01:** SmaRTClock Interface is locked. First key code (0xA5) has been written, waiting for second key code. **0x02:** SmaRTClock Interface is unlocked. First and second key codes (0xA5, 0xF1) have been written. **0x03:** SmaRTClock Interface is disabled until the next system reset. **Write:** When RTC0ST = 0x00 (locked), writing 0xA5 followed by 0xF1 unlocks the SmaRTClock Interface. When RTC0ST = 0x01 (waiting for second key code), writing any value other than the second key code (0xF1) will change RTC0STATE to 0x03 and disable the SmaRTClock Interface until the next system reset. When RTC0ST = 0x02 (unlocked), any write to RTC0KEY will lock the SmaRT-Clock Interface. When RTC0ST = 0x03 (disabled), writes to RTC0KEY have no effect.

SFR Definition 21.1. XBR0: Port I/O Crossbar Register 0

Bit	7	6	5	4	3	2	1	0
Name	CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xE1

Bit	Name	Function
7	CP1AE	**Comparator1 Asynchronous Output Enable.** 0: Asynchronous CP1 output unavailable at Port pin. 1: Asynchronous CP1 output routed to Port pin.
6	CP1E	**Comparator1 Output Enable.** 0: CP1 output unavailable at Port pin. 1: CP1 output routed to Port pin.
5	CP0AE	**Comparator0 Asynchronous Output Enable.** 0: Asynchronous CP0 output unavailable at Port pin. 1: Asynchronous CP0 output routed to Port pin.
4	CP0E	**Comparator0 Output Enable.** 0: CP1 output unavailable at Port pin. 1: CP1 output routed to Port pin.
3	SYSCKE	**$\overline{\text{SYSCLK}}$ Output Enable.** 0: $\overline{\text{SYSCLK}}$ output unavailable at Port pin. 1: $\overline{\text{SYSCLK}}$ output routed to Port pin.
2	SMB0E	**SMBus I/O Enable.** 0: SMBus I/O unavailable at Port pin. 1: SDA and SCL routed to Port pins.
1	SPI0E	**SPI0 I/O Enable.** 0: SPI0 I/O unavailable at Port pin. 1: SCK, MISO, and MOSI (for SPI0) routed to Port pins. NSS (for SPI0) routed to Port pin only if SPI0 is configured to 4-wire mode.
0	URT0E	**UART0 Output Enable.** 0: UART I/O unavailable at Port pin. 1: TX0 and RX0 routed to Port pins P0.4 and P0.5.
Note: SPI0 can be assigned either 3 or 4 Port I/O pins.		

22. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I^2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. The SMBus peripheral can be fully driven by software (i.e. software accepts/rejects slave addresses, and generates ACKs), or hardware slave address recognition and automatic ACK generation can be enabled to minimize software overhead. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 22.1.

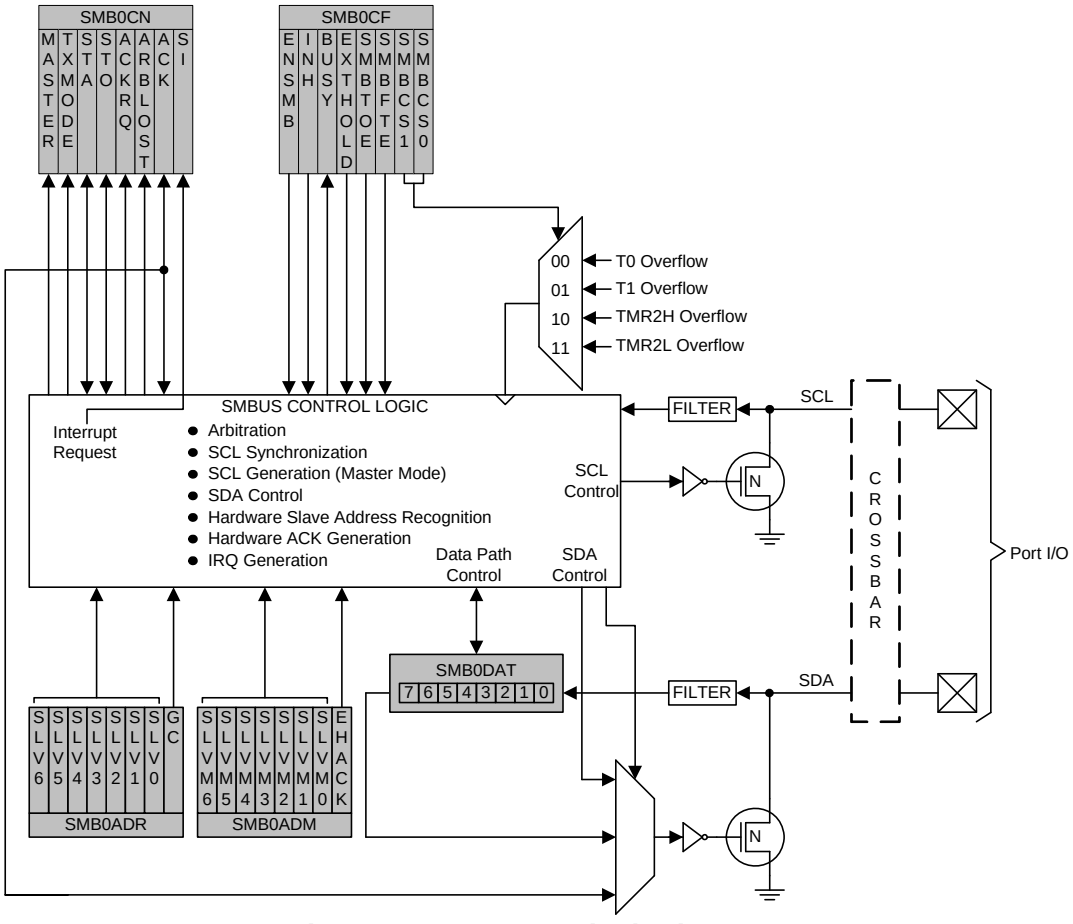

Figure 22.1. SMBus Block Diagram

SFR Definition 24.1. SPInCFG: SPI Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Addresses: SPI0CFG = 0xA1, SPI1CFG = 0x84
SFR Pages: SPI0CFG = 0x0, SPI1CFG = 0x0

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI Clock Phase.** 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	**SPI Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** Set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	**NSS Instantaneous Pin Input.** This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	**Shift Register Empty (valid in slave mode only).** Set to logic 1 when data has been transferred in/out of the shift register, and there is no data is available to read from the transmit buffer or write to the receive buffer. Set to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. Note: SRMT = 1 in Master Mode.
0	RXBMT	**Receive Buffer Empty (valid in slave mode only).** Set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. Note: RXBMT = 1 in Master Mode.

***Note:** In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 24.1 for timing parameters.

SFR Definition 24.2. SPInCN: SPI Control

Bit	7	6	5	4	3	2	1	0
Name	SPIFn	WCOLn	MODFn	RXOVRNn	NSSnMD1	NSSnMD0	TXBMTn	SPInEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	1	1	0

SFR Addresses: SPI0CN = 0xF8, Bit-Addressable; SPI1CN = 0xB0, Bit-Addressable
SFR Pages: SPI0CN = 0x0, SPI1CN = 0x0

Bit	Name	Function
7	SPIFn	**SPIn Interrupt Flag.** This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPIn interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.
6	WCOLn	**Write Collision Flag.** This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. It must be cleared by software.
5	MODFn	**Mode Fault Flag.** This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.
4	RXOVRNn	**Receive Overrun Flag (valid in slave mode only).** This bit is set to logic 1 by hardware (and generates a SPIn interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI shift register. This bit is not automatically cleared by hardware. It must be cleared by software.
3:2	NSSnMD[1:0]	**Slave Select Mode.** Selects between the following NSS operation modes: (See Section 24.2 and Section 24.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMTn	**Transmit Buffer Empty.** This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPInEN	**SPIn Enable.** 0: SPIn disabled. 1: SPIn enabled.

SFR Definition 24.3. SPInCKR: SPI Clock Rate

Bit	7	6	5	4	3	2	1	0
Name	SCRn[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Addresses: SPI0CKR = 0xA2, SPI1CKR = 0x85
SFR Pages: SPI0CKR = 0x0, SPI1CKR = 0x0

Bit	Name	Function
7:0	SCRn	**SPI Clock Rate.** These bits determine the frequency of the SCK output when the SPI module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPInCKR* is the 8-bit value held in the SPInCKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPInCKR[7:0] + 1)}$ for 0 <= SPI0CKR <= 255 Example: If SYSCLK = 2 MHz and SPInCKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 24.4. SPInDAT: SPI Data

Bit	7	6	5	4	3	2	1	0
Name	SPInDAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Addresses: SPI0DAT = 0xA3, SPI1DAT = 0x86
SFR Pages: SPI0DAT = 0x0, SPI1DAT = 0x0

Bit	Name	Function
7:0	SPInDAT	**SPIn Transmit and Receive Data.** The SPInDAT register is used to transmit and receive SPIn data. Writing data to SPInDAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPInDAT returns the contents of the receive buffer.

27. C2 Interface

C8051F91x-C8051F90x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

27.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 27.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function	
7:0	C2ADD[7:0]	**C2 Address.** The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.	
		Address	**Description**
		0x00	Selects the Device ID register for Data Read instructions
		0x01	Selects the Revision ID register for Data Read instructions
		0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
		0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions