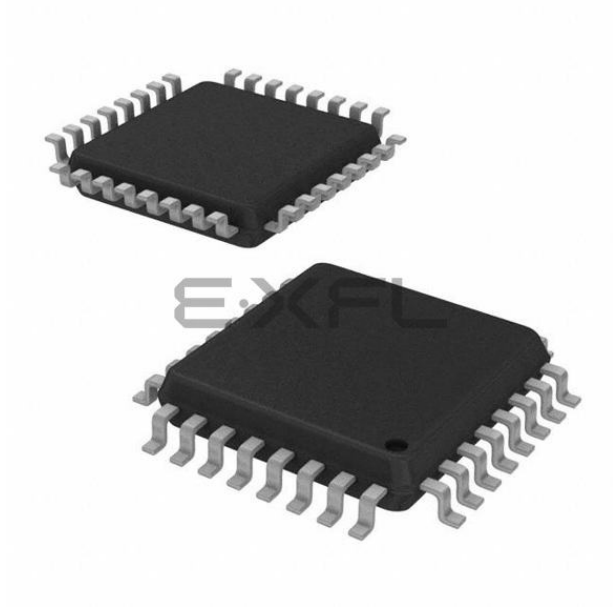

Welcome to E-XFL.COM

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	Discontinued at Digi-Key
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	29
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 21x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f366-gq

cated by a status bit and an interrupt (if enabled). The resulting 10-bit data word is latched into the ADC data SFRs upon completion of a conversion.

Window compare registers for the ADC data can be configured to interrupt the controller when ADC data is either within or outside of a specified range. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within/outside the specified range.

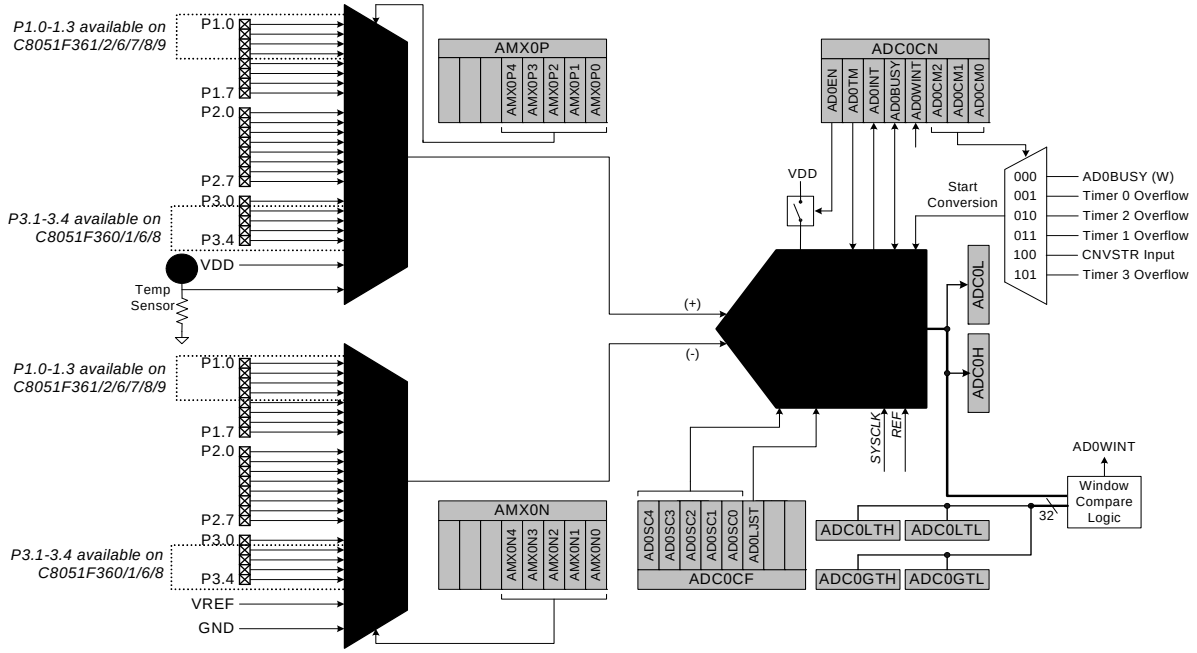

Figure 1.11. 10-Bit ADC Block Diagram

1.8. Comparators

C8051F36x devices include two on-chip voltage comparators that are enabled/disabled and configured via user software. Port I/O pins may be configured as comparator inputs via a selection mux. Two comparator outputs may be routed to a Port pin if desired: a latched output and/or an unlatched (asynchronous) output. Comparator response time is programmable, allowing the user to select between high-speed and low-power modes. Positive and negative hysteresis are also configurable.

Comparator interrupts may be generated on rising, falling, or both edges. When in IDLE mode, these interrupts may be used as a “wake-up” source. Comparator0 may also be configured as a reset source. Figure 1.12 shows the Comparator0 block diagram, and Figure 1.13 shows the Comparator1 block diagram.

Note: The first Port I/O pins shown in Figure 1.12 and Figure 1.13 are for the 48-pin (C8051F360/3) devices. The second set of Port I/O pins are for the 32-pin and 28-pin (C8051F361/2/4/5/6/7/8/9) devices. Please refer to the CPTnMX registers (SFR Definition 8.2 and SFR Definition 8.5) for more information.

4. Pinout and Package Definitions

Table 4.1. Pin Definitions for the C8051F36x

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
V_{DD}	19, 31, 43	4	4		Power Supply Voltage.
GND	18, 30, 42	3	3		Ground.
AGND	6	—	—		Analog Ground.
AV+	7	—	—		Analog Supply Voltage. Must be tied to +2.7 to +3.6 V.
$\overline{RST}$/	8	5	5	D I/O	Device Reset. Open-drain output of internal POR or V_{DD} Monitor. An external source can initiate a system reset by driving this pin low for at least 10 µs.
C2CK				D I/O	Clock signal for the C2 Debug Interface.
P4.6/	9	—	—	D I/O or A In	Port 4.6. See Section 17 for a complete description.
C2D				D I/O	Bi-directional data signal for the C2 Debug Interface.
P3.0/	—	6	6	D I/O or A In	Port 3.0. See Section 17 for a complete description.
C2D				D I/O	Bi-directional data signal for the C2 Debug Interface.
P0.0	5	2	2	D I/O or A In	Port 0.0. See Section 17 for a complete description.
P0.1	4	1	1	D I/O or A In	Port 0.1. See Section 17 for a complete description.
P0.2	3	32	28	D I/O or A In	Port 0.2. See Section 17 for a complete description.
P0.3	2	31	27	D I/O or A In	Port 0.3. See Section 17 for a complete description.
P0.4	1	30	26	D I/O or A In	Port 0.4. See Section 17 for a complete description.
P0.5	48	29	25	D I/O or A In	Port 0.5. See Section 17 for a complete description.
P0.6	47	28	24	D I/O or A In	Port 0.6. See Section 17 for a complete description.
P0.7	46	27	23	D I/O or A In	Port 0.7. See Section 17 for a complete description.

Table 4.1. Pin Definitions for the C8051F36x (Continued)

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
P3.1	24	7	—	D I/O or A In	Port 3.1. See Section 17 for a complete description.
P3.2	23	8	—	D I/O or A In	Port 3.2. See Section 17 for a complete description.
P3.3	22	9	—	D I/O or A In	Port 3.3. See Section 17 for a complete description.
P3.4	21	10	—	D I/O or A In	Port 3.4. See Section 17 for a complete description.
P3.5	20	—	—	D I/O or A In	Port 3.5. See Section 17 for a complete description.
P3.6	17	—	—	D I/O or A In	Port 3.6. See Section 17 for a complete description.
P3.7	16	—	—	D I/O or A In	Port 3.7. See Section 17 for a complete description.
P4.0	15	—	—	D I/O or A In	Port 4.0. See Section 17 for a complete description.
P4.1	14	—	—	D I/O	Port 4.1. See Section 17 for a complete description.
P4.2	13	—	—	D I/O	Port 4.2. See Section 17 for a complete description.
P4.3	12	—	—	D I/O	Port 4.3. See Section 17 for a complete description.
P4.4	11	—	—	D I/O	Port 4.4. See Section 17 for a complete description.
P4.5	10	—	—	D I/O	Port 4.5. See Section 17 for a complete description.

On exit from the PCA interrupt service routine, the CIP-51 will return to the ADC0 Window Comparator ISR. On execution of the RETI instruction, SFR Page 0x00 used to access the PCA registers will be automatically popped off of the SFR Page Stack, and the contents of the SFRNEXT register will be moved to the SFRPAGE register. Software in the ADC0 ISR can continue to access SFR's as it did prior to the PCA interrupt. Likewise, the contents of SFRLAST are moved to the SFRNEXT register. Recall this was the SFR Page value 0x0F being used to access OSCICN before the ADC0 interrupt occurred. See Figure 9.7 below.

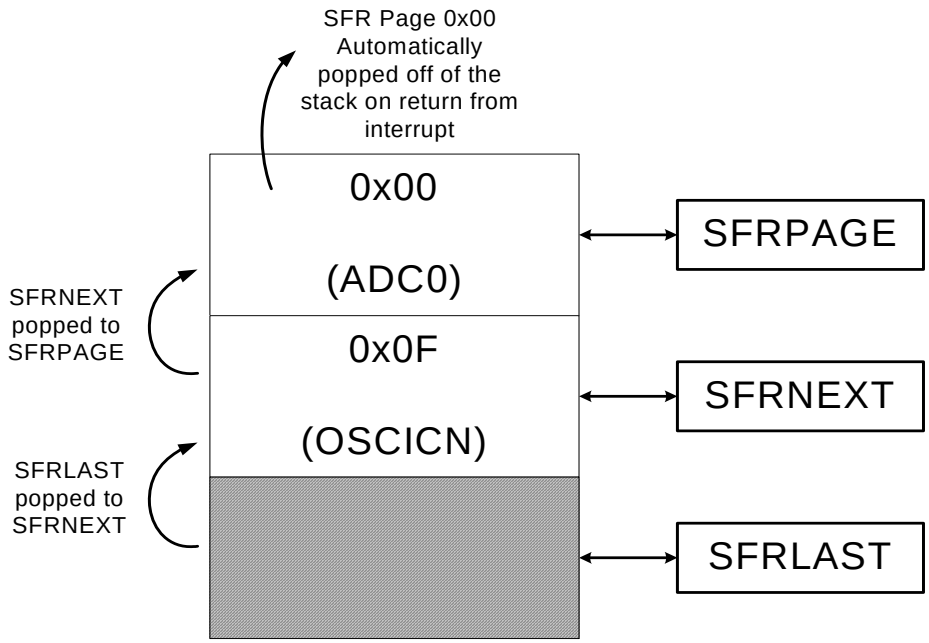

Figure 9.7. SFR Page Stack Upon Return From PCA Interrupt

11. Multiply And Accumulate (MAC0)

The C8051F36x devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. MAC0 contains a 16-by-16 bit multiplier and a 40-bit adder, which can perform integer or fractional multiply-accumulate and multiply operations on signed input values in two SYSCLK cycles. A rounding engine provides a rounded 16-bit fractional result after an additional (third) SYSCLK cycle. MAC0 also contains a 1-bit arithmetic shifter that will left or right-shift the contents of the 40-bit accumulator in a single SYSCLK cycle. Figure 11.1 shows a block diagram of the MAC0 unit and its associated Special Function Registers.

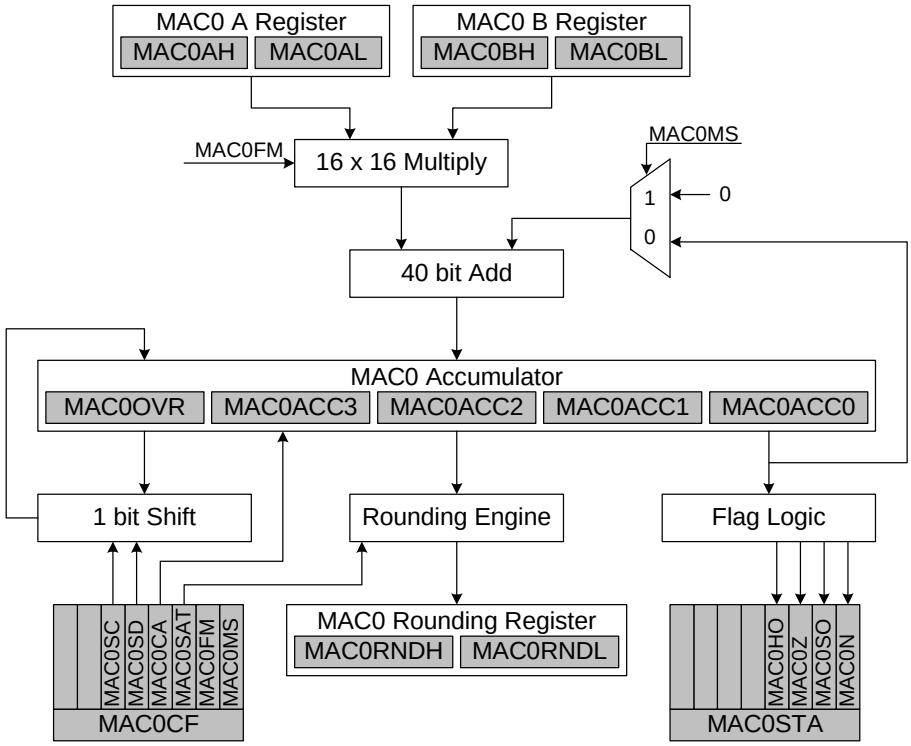

Figure 11.1. MAC0 Block Diagram

11.1. Special Function Registers

There are thirteen Special Function Register (SFR) locations associated with MAC0. Two of these registers are related to configuration and operation, while the other eleven are used to store multi-byte input and output data for MAC0. The Configuration register MAC0CF (SFR Definition 11.1) is used to configure and control MAC0. The Status register MAC0STA (SFR Definition 11.2) contains flags to indicate overflow conditions, as well as zero and negative results. The 16-bit MAC0A (MAC0AH:MAC0AL) and MAC0B (MAC0BH:MAC0BL) registers are used as inputs to the multiplier. The MAC0 Accumulator register is 40 bits long, and consists of five SFRs: MAC0OVR, MAC0ACC3, MAC0ACC2, MAC0ACC1, and MAC0ACC0. The primary results of a MAC0 operation are stored in the Accumulator registers. If they are needed, the rounded results are stored in the 16-bit Rounding Register MAC0RND (MAC0RNDH:MAC0RNDL).

SFR Definition 11.13. MAC0RNDL: MAC0 Rounding Register Low Byte

SFR Page: 0
SFR Address: 0xAE

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Low Byte (bits 7–0) of MAC0 Rounding Register.

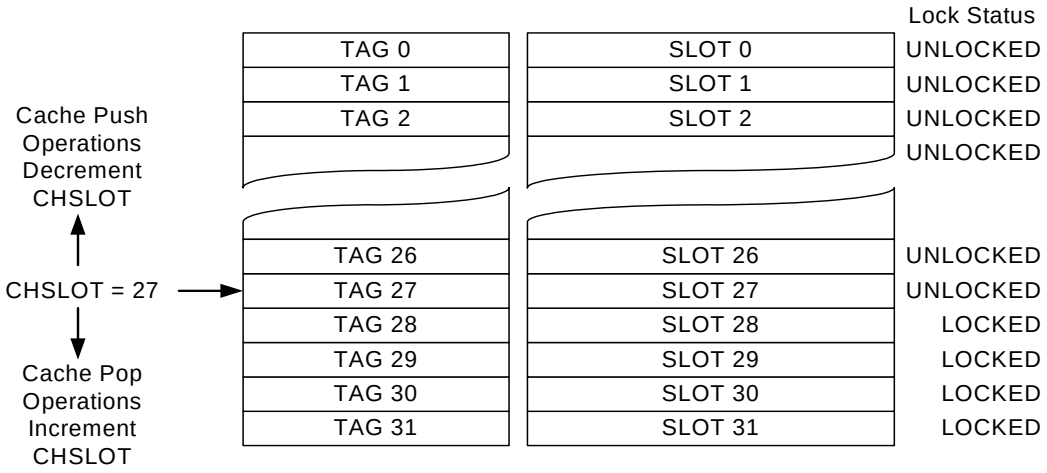

Figure 14.3. Cache Lock Operation

16.3. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 16.1. A 10 MΩ resistor also must be wired across the XTAL1 and XTAL2 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 16.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 16.5).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.5 and P0.6 (C8051F360/3) or P0.2 and P0.3 (C8051F361/2/4/5/6/7/8/9) are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.6 (C8051F360/3) or P0.3 (C8051F361/2/4/5/6/7/8/9) is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “17.1. Priority Crossbar Decoder” on page 184 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “17.2. Port I/O Initialization” on page 186 for details on Port input mode selection.

16.4. System Clock Selection

The internal oscillator requires little start-up time, and may be enabled and selected as the system clock in the same write to OSCICN. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use as the system clock. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to ‘1’ by hardware when the external oscillator is settled. To avoid reading a false XTLVLD, in crystal mode software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD. RC and C modes typically require no startup time. The PLL also requires time to lock onto the desired frequency, and the PLL Lock Flag (PLLLCK in register PLL0CN) is set to ‘1’ by hardware once the PLL is locked on the correct frequency.

The CLKSL1-0 bits in register CLKSEL select which oscillator source generates the system clock. CLKSL1-0 must be set to ‘01’ for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals, such as the timers and PCA, when the internal oscillator or the PLL is selected as the system clock. The system clock may be switched on-the-fly between the internal and external oscillators or the PLL, so long as the selected oscillator source is enabled and settled.

16.5. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 16.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 16.5 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is enabled, the oscillator amplitude detection circuit requires a settle time to achieve proper bias. Waiting at least 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Force the XTAL1 and XTAL2 pins low by writing 0's to the port latch.
Step 2. Configure XTAL1 and XTAL2 as analog inputs.
Step 3. Enable the external oscillator.
Step 4. Wait at least 1 ms.
Step 5. Poll for XTLVLD => '1'.
Step 6. Switch the system clock to the external oscillator.

Note: Tuning-fork crystals may require additional settling time before XTLVLD returns a valid result.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 16.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 16.2.

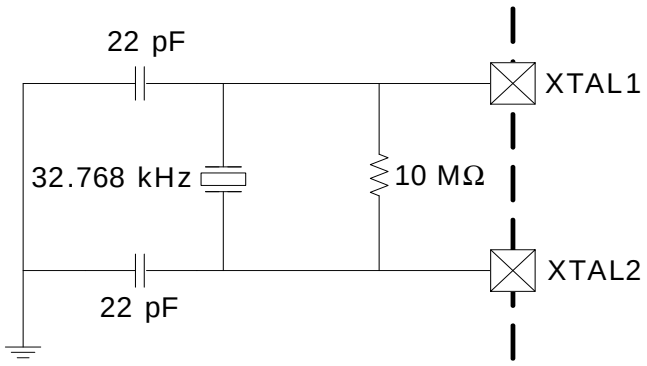

Figure 16.2. 32.768 kHz External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

16.8.3. Powering on and Initializing the PLL

To set up and use the PLL as the system clock after power-up of the device, the following procedure should be implemented:

Step 1. Ensure that the reference clock to be used (internal or external) is running and stable.
Step 2. Set the PLLSRC bit (PLL0CN.2) to select the desired clock source for the PLL.
Step 3. Program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see Section “13. Flash Memory” on page 135).
Step 4. Enable power to the PLL by setting PLLPWR (PLL0CN.0) to ‘1’.
Step 5. Program the PLL0DIV register to produce the divided reference frequency to the PLL.
Step 6. Program the PLLLP3–0 bits (PLL0FLT.3–0) to the appropriate range for the divided reference frequency.
Step 7. Program the PLLICO1–0 bits (PLL0FLT.5–4) to the appropriate range for the PLL output frequency.
Step 8. Program the PLL0MUL register to the desired clock multiplication factor.
Step 9. Wait at least 5 µs, to provide a fast frequency lock.
Step 10. Enable the PLL by setting PLLEN (PLL0CN.1) to ‘1’.
Step 11. Poll PLLLCK (PLL0CN.4) until it changes from ‘0’ to ‘1’.
Step 12. Switch the System Clock source to the PLL using the CLKSEL register.

If the PLL characteristics need to be changed when the PLL is already running, the following procedure should be implemented:

Step 1. The system clock should first be switched to either the internal oscillator or an external clock source that is running and stable, using the CLKSEL register.
Step 2. Ensure that the reference clock to be used for the new PLL setting (internal or external) is running and stable.
Step 3. Set the PLLSRC bit (PLL0CN.2) to select the new clock source for the PLL.
Step 4. If moving to a faster frequency, program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see Section “13. Flash Memory” on page 135).
Step 5. Disable the PLL by setting PLLEN (PLL0CN.1) to ‘0’.
Step 6. Program the PLL0DIV register to produce the divided reference frequency to the PLL.
Step 7. Program the PLLLP3–0 bits (PLL0FLT.3–0) to the appropriate range for the divided reference frequency.
Step 8. Program the PLLICO1-0 bits (PLL0FLT.5–4) to the appropriate range for the PLL output frequency.
Step 9. Program the PLL0MUL register to the desired clock multiplication factor.
Step 10. Enable the PLL by setting PLLEN (PLL0CN.1) to ‘1’.
Step 11. Poll PLLLCK (PLL0CN.4) until it changes from ‘0’ to ‘1’.
Step 12. Switch the System Clock source to the PLL using the CLKSEL register.
Step 13. If moving to a slower frequency, program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see Section “13. Flash Memory” on page 135). **Important Note: Cache reads, cache writes, and the prefetch engine should be disabled whenever the FLRT bits are changed to a lower setting.**

SFR Definition 17.9. P1: Port1

SFR Page: **all pages**
SFR Address: **0x90** (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P1.[7:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P1MDOUT.n bit = 0).
Read - Always reads '0' if selected as analog input in register P1MDIN. Directly reads Port pin when configured as digital input.
0: P1.n pin is logic low.
1: P1.n pin is logic high.

SFR Definition 17.10. P1MDIN: Port1 Input Mode

SFR Page: **F**
SFR Address: **0xF2**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Analog Input Configuration Bits for P1.7-P1.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P1.n pin is configured as an analog input.
1: Corresponding P1.n pin is not configured as an analog input.

18.4.3. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic ‘0’, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 18.3. SMB0DAT: SMBus Data

SFR Page: all pages
SFR Address: 0xC2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SMB0DAT: SMBus Data.
The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic ‘1’. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

20.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic '1' at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 20.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 20.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 20.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

SFR Definition 20.3. SPI0CKR: SPI0 Clock Rate

SFR Page: all pages
SFR Address: 0xA2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SCR7–SCR0: SPI0 Clock Rate.
These bits determine the frequency of the SCK output when the SPI0 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 *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

SFR Definition 20.4. SPI0DAT: SPI0 Data

SFR Page: all pages
SFR Address: 0xA3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

SFR Definition 21.2. TMOD: Timer Mode

SFR Page: all pages
SFR Address: 0x89

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 10.7).

Bit 6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits 5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit 3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 10.7).

Bit 2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits 1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

SFR Definition 21.13. TMR3CN: Timer 3 Control

SFR Page: all pages
SFR Address: 0x91

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3	–	T3XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: TF3H: Timer 3 High Byte Overflow Flag.
Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. TF3H is not automatically cleared by hardware and must be cleared by software.

Bit 6: TF3L: Timer 3 Low Byte Overflow Flag.
Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF3LEN is set and Timer 3 interrupts are enabled. TF3L will set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.

Bit 5: TF3LEN: Timer 3 Low Byte Interrupt Enable.
This bit enables/disables Timer 3 Low Byte interrupts. If TF3LEN is set and Timer 3 interrupts are enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
0: Timer 3 Low Byte interrupts disabled.
1: Timer 3 Low Byte interrupts enabled.

Bit 4: TF3CEN: Timer 3 Low-Frequency Oscillator Capture Enable.
This bit enables/disables Timer 3 Low-Frequency Oscillator Capture Mode. If TF3CEN is set and Timer 3 interrupts are enabled, an interrupt will be generated on a rising edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR3H:TMR3L will be copied to TMR3RLH:TMR3RLL. See Section “16. Oscillators” on page 168 for more details.
0: Timer 3 Low-Frequency Oscillator Capture disabled.
1: Timer 3 Low-Frequency Oscillator Capture enabled.

Bit 3: T3SPLIT: Timer 3 Split Mode Enable.
When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload.
0: Timer 3 operates in 16-bit auto-reload mode.
1: Timer 3 operates as two 8-bit auto-reload timers.

Bit 2: TR3: Timer 3 Run Control.
This bit enables/disables Timer 3. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in this mode.
0: Timer 3 disabled.
1: Timer 3 enabled.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: T3XCLK: Timer 3 External Clock Select.
This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.
0: Timer 3 external clock selection is the system clock divided by 12.
1: Timer 3 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

bit contained a ‘0’ prior to the execution of the “CPL C” opcode, it will properly transition to a 1 when the execution phase of the opcode has completed. This is illustrated in the following table:

Correct operation	Correct operation	Failure case
initial state of C is 1	initial state of C is 0	initial state of C is 1
CPL C	CPL C	CPL C
final state of C is 0	final state of C is 1	final state of C is 1

The instruction order dependency is as follows:

In the failure case, the CPL C opcode must be immediately preceded by a JB, JNB, or JBC opcode.

JB, JNB, and JBC are all conditional branch instructions (JB is “Jump if bit is set”, JNB is “Jump if bit is not set”, and JBC is “Jump if bit is set and clear bit”). Because the branches are conditional, they have both a “branch taken” condition as well as a “branch not taken” condition. Both “branch taken” and “branch not taken” conditions may exhibit the error, as long as the CPL C opcode executes immediately after the branch instruction has executed.

Impacts

The CPL C opcode is often used in math operations, such as address calculations for pointer arithmetic. If present, this behavior can cause undesirable and unpredictable program execution.

The occurrence of this behavior is sensitive to system clock frequency, temperature, and power supply voltage as follows:

JB / JNB / JBC + CPL C opcode sequence present?	VDD	System clock frequency	Temperature range	Failure possible?
No	≥ 3.0 V	≤ 100 MHz	-40 to +85 °C	No
Yes	≥ 3.0 V	≤ 70 MHz	-40 to +85 °C	No

Workaround

The bug can best be addressed by checking to see if the problematic instruction sequence is present in the device firmware and removing it if detected. In most cases, the firmware can be changed to insert a NOP instruction immediately before the CPL C opcode, so that the CPL C instruction does not immediately follow the JB / JNB / JBC opcode in the code execution path.

Silicon Labs has developed a hex file scanner that can be used to determine if a code project contains the instruction sequence above. Instructions for using the scanner, as well as details regarding the scanner’s operation can be found here:

http://community.silabs.com/t5/Silicon-Labs-Knowledge-Base/C8051F360-Rev-B-Erratum-CPU-E101-CPL-C-HEX-Scanner/ta-p/133808

This behavior has been corrected on Revision C of this device.

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com