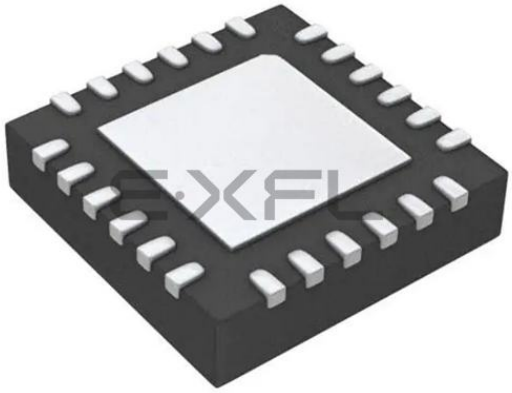

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	CIP-51™
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C/SMBus, I²C Slave, SPI, UART/USART
Peripherals	CapSense, DMA, POR, PWM, WDT
Number of I/O	19
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f972-a-gmr

2.1. Power

2.1.1. Voltage Supply Monitor (VMON0)

The C8051F97x devices include a voltage supply monitor which allows devices to function in known, safe operating condition without the need for external hardware.

The supply monitor module includes the following features:

- Holds the device in reset if the main VDD supply drops below the VDD Reset threshold.

2.1.2. Device Power Modes

The C8051F97x devices feature seven low power modes in addition to normal operating mode, allowing the designer to save power when the core is not in use. All power modes are detailed in Table 2.1.

Table 2.1. Power Modes

Power Mode	Description	Wake-up Sources	Power and Performance
Normal Active	Device fully functional.	N/A	Excellent MIPS/mW
Low Power Active	Device fully functional except peripherals whose clocks are intentionally disabled.	N/A	Excellent. Clocks only enabled for peripherals that request for it.
Idle	All peripherals fully functional. Wake-up in 2 clock cycles.	Any Interrupt	Good No Code Execution
Low Power Idle	Similar to Idle mode, the CPU is halted. Clocks of unused peripherals can be intentionally gated. Wake-up in 2 clock cycles.	Any Interrupt	Very Good No Code Execution. Clocks only enabled for peripherals that request for it.
Stop	Legacy 8051 low power mode. A reset is required to wake up.	Any Reset	Good No Code Execution Precision Oscillator Disabled
Suspend	Similar to Stop mode, but very fast wake-up time and code resumes execution at the next instruction.	CS0, SmaRTClock, Port Match, I^2C Slave, $\overline{RST}$ pin	Very Good No Code Execution All Internal Oscillators Disabled System Clock Gated
Sleep	Ultra Low Power and flexible wake-up sources. Code resumes execution at the next instruction.	SmaRTClock, Port Match, I^2C Slave, $\overline{RST}$ pin	Excellent Power Supply Gated All Oscillators except SmaRTClock Disabled

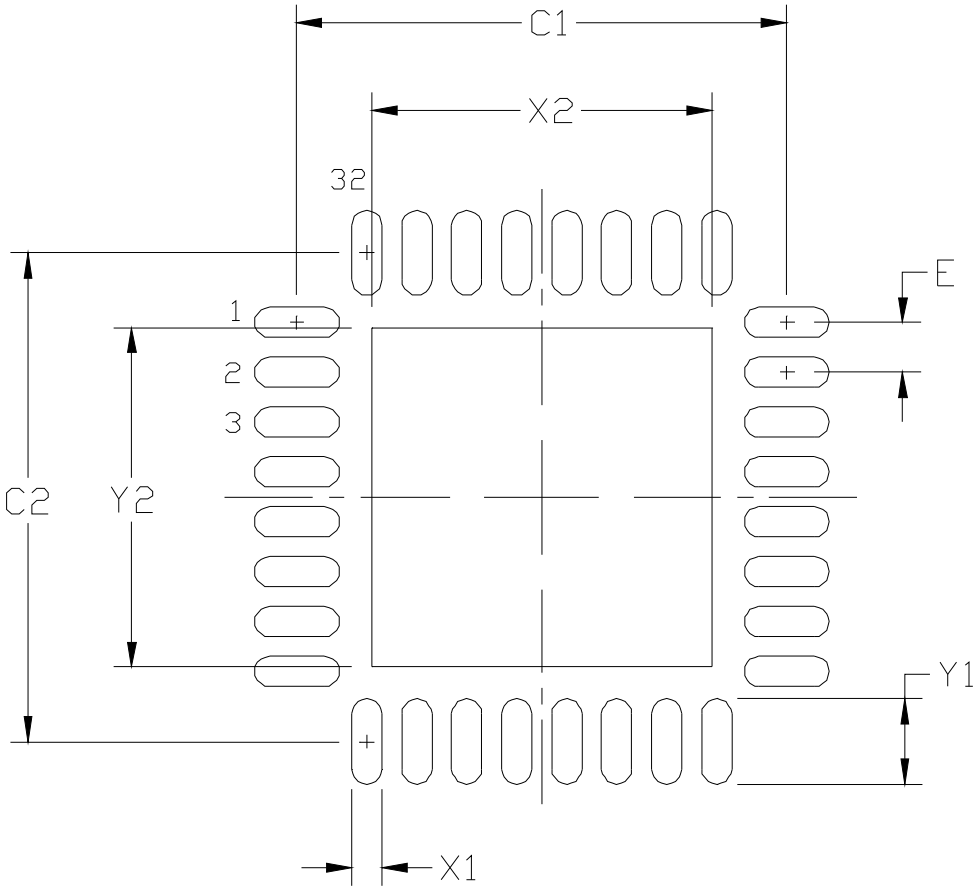

Figure 6.2. Typical QFN-32 Landing Diagram

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
TL1	0x8B	0x0	Timer 1 Low Byte	405
TMOD	0x89	0x0	Timer 0/1 Mode	403
TMR2CN	0xC8	All pages	Timer 2 Control	408
TMR2H	0xCD	0x0	Timer 2 High Byte	413
TMR2L	0xCC	0x0	Timer 2 Low Byte	412
TMR2RLH	0xCB	0x0	Timer 2 Reload High Byte	411
TMR2RLL	0xCA	0x0	Timer 2 Reload Low Byte	410
TMR3CN	0x91	0x0	Timer 3 Control	414
TMR3H	0x95	0x0	Timer 3 High Byte	419
TMR3L	0x94	0x0	Timer 3 Low Byte	418
TMR3RLH	0x93	0x0	Timer 3 Reload High Byte	417
TMR3RLL	0x92	0x0	Timer 3 Reload Low Byte	416
TOFFH	0x8E	0xF	Temperature Sensor Offset High	128
TOFFL	0x8D	0xF	Temperature Sensor Offset Low	129
VDM0CN	0xFF	0x0	Supply Monitor Control	327
XBR0	0x95	0xF	Port I/O Crossbar 0	285
XBR1	0x96	0xF	Port I/O Crossbar 1	286

10.2. Programming the Flash Memory

Writes to flash memory clear bits from logic 1 to logic 0, and can be performed on single byte locations. Flash erasures set bits back to logic 1, and occur only on full pages. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a flash write/erase operation.

The simplest means of programming the flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device.

To ensure the integrity of flash contents, it is strongly recommended that the on-chip supply monitor be enabled in any system that includes code that writes and/or erases flash memory from software.

10.2.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a flash write or erase is attempted before the key codes have been written properly. The flash lock resets after each write or erase; the key codes must be written again before a following flash operation can be performed.

10.2.2. Flash Erase Procedure

The flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to flash memory using MOVX, flash write operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit in the PSCTL register to logic 1 (this directs the MOVX writes to target flash memory); and (2) Writing the flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

A write to flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in flash. **A byte location to be programmed should be erased before a new value is written.** Erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire page, perform the following steps:

1. Disable interrupts (recommended).
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. Write the first key code to FLKEY: 0xA5.
5. Write the second key code to FLKEY: 0xF1.
6. Using the MOVX instruction, write a data byte to any location within the page to be erased.
7. Clear the PSWE and PSEE bits.

10.2.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the flash page containing the target location, as described in Section 10.2.2.
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the desired page.
8. Clear the PSWE bit.

Steps 5–7 must be repeated for each byte to be written. After flash writes are complete, PSWE should be cleared so that MOVX instructions do not target program memory.

Register 12.2. REVID: Revision Identifcation

Bit	7	6	5	4	3	2	1	0
Name	REVID							
Type	R							
Reset	X	X	X	X	X	X	X	X
SFR Page = 0xF; SFR Address: 0xE2								

Table 12.3. REVID Register Bit Descriptions

Bit	Name	Function
7:0	REVID	**Revision ID.** This read-only register returns the 8-bit revision ID. 00000000: Reserved. 00000001: Revision A 00000010-11111111: Reserved.

Table 13.4. EIE1 Register Bit Descriptions

Bit	Name	Function
0	ESMB0	**SMBus (SMB0) Interrupt Enable.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

On exit from the PCA interrupt service routine, the CIP-51 will return to the SPI0 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. Firmware in the SPI0 ISR can continue to access SFRs 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 I2C0STAT before the SPI0 interrupt occurred. See Figure 20.6.

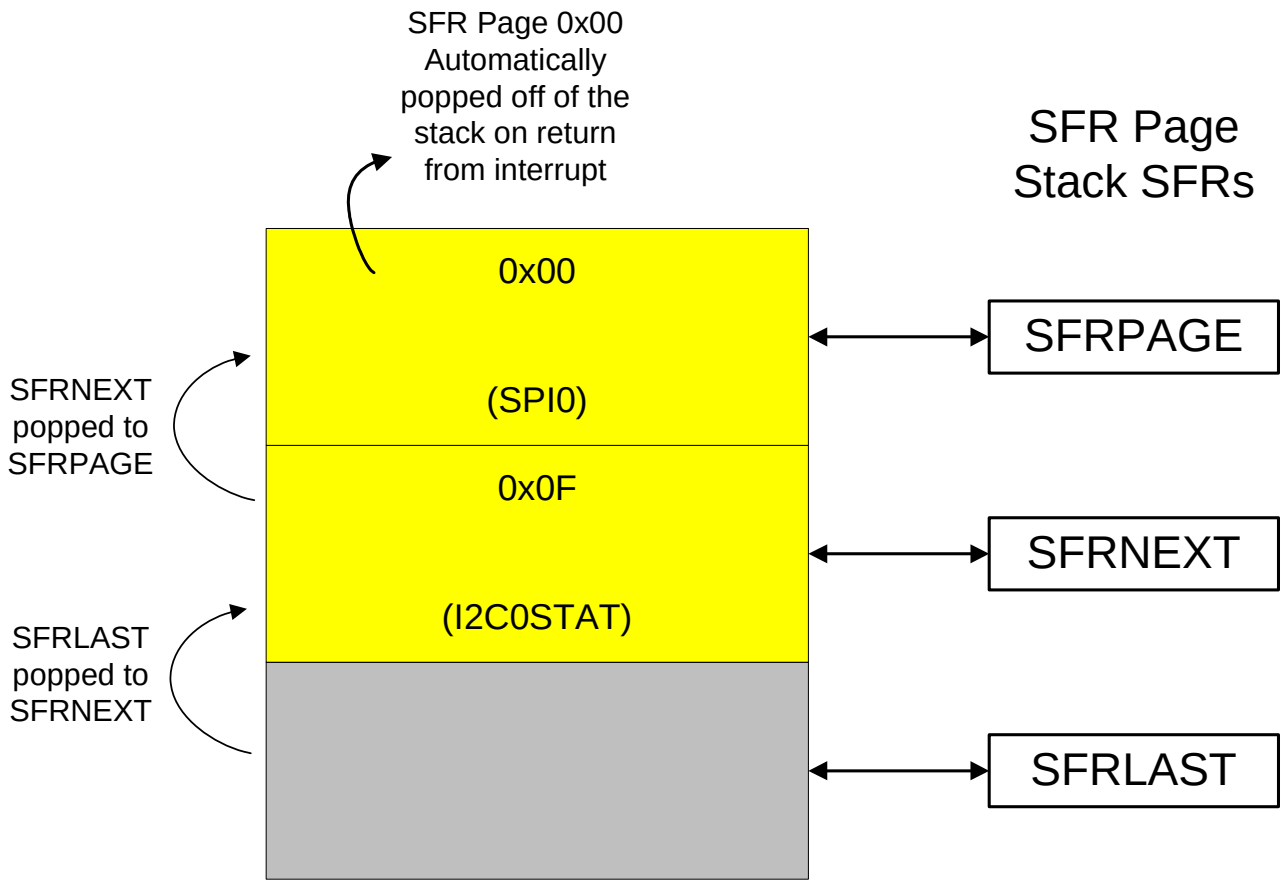

Figure 20.6. SFR Page Stack Upon Return From PCA Interrupt

Register 21.9. DMA0NAOH: Memory Address Offset High

Bit	7	6	5	4	3	2	1	0
Name	Reserved						NAOH	
Type	R						RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCC								

Bit	Name	Function
7:2	Reserved	Must write reset value.
1:0	NAOH	**Memory Address Offset High.** This field is the high byte of the channel offset address. The base address added to the offset address creates the current channel XRAM address. The address offset auto-increments by one after one byte is transferred. When configuring a channel for a DMA transfer, the address offset should be reset to 0.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.12. DMA0NSZL: Memory Transfer Size Low

Bit	7	6	5	4	3	2	1	0
Name	NSZL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCD								

Bit	Name	Function
7:0	NSZL	**Memory Transfer Size Low.** This field sets the low byte of the number of DMA transfers for the selected channel.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 24.3. CLKMODE: Clock Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved					LPME	ECSR	FCAM
Type	R					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xF7								

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	LPME	**Low Power Mode Enable.** Setting this bit allows the device to enter low power active/idle mode. PCLKEN settings are only enabled when this bit is set.
1	ECSR	**Clock Request Enable.** When this bit is set, the source clocks will only be requested when an incoming request is active or a peripheral clock is enabled.
0	FCAM	**Force Clock Tree Enable.** When this bit is set, clock gating will only be performed in idle mode. This allows a user to have complete access to all SFRs in active mode, but still get the benefits of clock gating in idle mode.

Register 24.5. OSCICN: High Frequency Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY	Reserved					
Type	RW	R	RW					
Reset	0	0	X	X	X	X	X	X
SFR Page = 0x0; SFR Address: 0xB2								

Bit	Name	Function
7	IOSCEN	**High Frequency Oscillator Enable.** 0: High Frequency Oscillator disabled. 1: High Frequency Oscillator enabled.
6	IFRDY	**Internal Oscillator Frequency Ready Flag.** 0: High Frequency Oscillator is not running at its programmed frequency. 1: High Frequency Oscillator is running at its programmed frequency.
5:0	Reserved	Must write reset value.

Notes:
1. Read-modify-write operations such as ORL and ANL must be used to set or clear the enable bit of this register.
2. OSCBIAS (REG0CN.4) must be set to 1 before enabling the High Frequency Oscillator.

Register 25.7. CAPTURE1: RTC Timer Capture 1

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x01								

Bit	Name	Function
7:0	CAPTURE1	**RTC Timer Capture 1.** The CAPTURE3-CAPTURE0 registers are used to read or set the 32-bit RTC timer. Data is transferred to or from the RTC timer when the RTC0SET or RTC0CAP bits are set.

Register 25.13. ALARM3: RTC Alarm Programmed Value 3

Bit	7	6	5	4	3	2	1	0
Name	ALARM3							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x0B								

Bit	Name	Function
7:0	ALARM3	**RTC Alarm Programmed Value 3.** The ALARM3-ALARM0 registers are used to set an alarm event for the RTC timer. The RTC alarm should be disabled (RTC0AEN=0) when updating these registers.

Registers XBR0 and XBR1are used to assign the digital I/O resources to the physical I/O port pins. Note that when the SMBus is selected, the crossbar assigns both pins associated with the SMBus (SDA and SCL); when UART0 is selected, the crossbar assigns both pins associated with UART0 (TX and RX). Standard port I/Os appear contiguously after the prioritized functions have been assigned.

Figure 26.3 shows an example of the resulting pin assignments of the device with SMBus0, SPI0, and two channels of PCA0 enabled and P0.3, P0.4, P1.0, and P1.1 skipped (P0SKIP = 0x18, P1SKIP = 0x03). SMBus0 is the highest priority and it will be assigned first. The next-highest enabled peripheral is SPI0. P0.2 is available, so SPI0 takes this pin. The next pins, MISO, MOSI, and NSS are routed to P0.5, P0.6, and P0.7, respectively, because P0.3 and P0.4 are skipped. The PCA0 CEX0 and CEX1are then routed to P1.2 and P1.3. The other pins on the device are available for use as general-purpose digital I/O or analog functions.

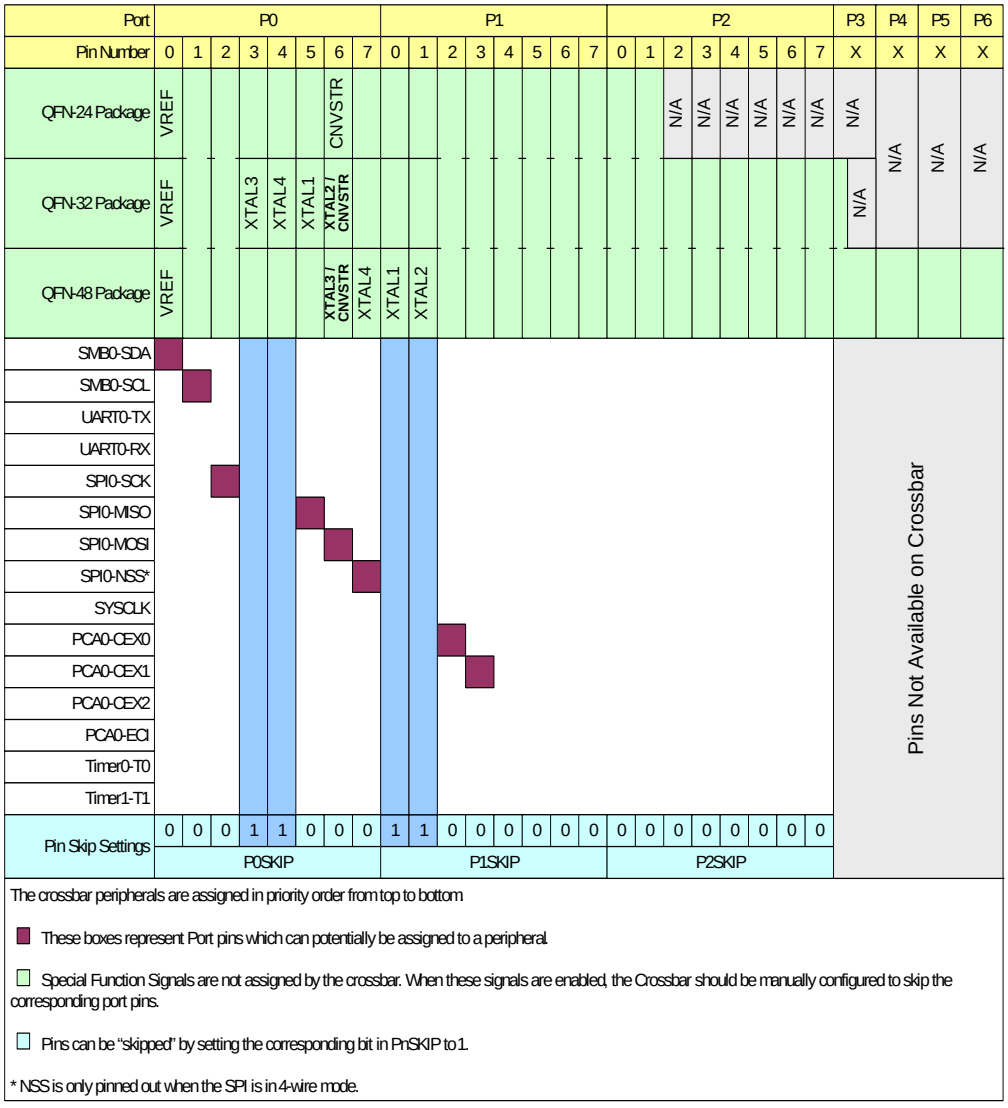

Figure 26.3. Crossbar Priority Decoder Example

Note: The SPI can be operated in either 3-wire or 4-wire modes, pending the state of the NSSMD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a port pin.

Register 26.13. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xED								

Table 26.16. P1MDIN Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Input Mode.** 0: P1.7 pin is configured for analog mode. 1: P1.7 pin is configured for digital mode.
6	B6	**Port 1 Bit 6 Input Mode.** 0: P1.6 pin is configured for analog mode. 1: P1.6 pin is configured for digital mode.
5	B5	**Port 1 Bit 5 Input Mode.** 0: P1.5 pin is configured for analog mode. 1: P1.5 pin is configured for digital mode.
4	B4	**Port 1 Bit 4 Input Mode.** 0: P1.4 pin is configured for analog mode. 1: P1.4 pin is configured for digital mode.
3	B3	**Port 1 Bit 3 Input Mode.** 0: P1.3 pin is configured for analog mode. 1: P1.3 pin is configured for digital mode.
2	B2	**Port 1 Bit 2 Input Mode.** 0: P1.2 pin is configured for analog mode. 1: P1.2 pin is configured for digital mode.
1	B1	**Port 1 Bit 1 Input Mode.** 0: P1.1 pin is configured for analog mode. 1: P1.1 pin is configured for digital mode.
0	B0	**Port 1 Bit 0 Input Mode.** 0: P1.0 pin is configured for analog mode. 1: P1.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.28. P4: Port 4 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0x0; SFR Address: 0xE2								

Table 26.31. P4 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Latch.** 0: P4.7 is low. Set P4.7 to drive low. 1: P4.7 is high. Set P4.7 to drive or float high.
6	B6	**Port 4 Bit 6 Latch.** 0: P4.6 is low. Set P4.6 to drive low. 1: P4.6 is high. Set P4.6 to drive or float high.
5	B5	**Port 4 Bit 5 Latch.** 0: P4.5 is low. Set P4.5 to drive low. 1: P4.5 is high. Set P4.5 to drive or float high.
4	B4	**Port 4 Bit 4 Latch.** 0: P4.4 is low. Set P4.4 to drive low. 1: P4.4 is high. Set P4.4 to drive or float high.
3	B3	**Port 4 Bit 3 Latch.** 0: P4.3 is low. Set P4.3 to drive low. 1: P4.3 is high. Set P4.3 to drive or float high.
2	B2	**Port 4 Bit 2 Latch.** 0: P4.2 is low. Set P4.2 to drive low. 1: P4.2 is high. Set P4.2 to drive or float high.
1	B1	**Port 4 Bit 1 Latch.** 0: P4.1 is low. Set P4.1 to drive low. 1: P4.1 is high. Set P4.1 to drive or float high.
0	B0	**Port 4 Bit 0 Latch.** 0: P4.0 is low. Set P4.0 to drive low. 1: P4.0 is high. Set P4.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.30. P4MDOUT: Port 4 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC3								

Table 26.33. P4MDOUT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Output Mode.** 0: P4.7 output is open-drain. 1: P4.7 output is push-pull.
6	B6	**Port 4 Bit 6 Output Mode.** 0: P4.6 output is open-drain. 1: P4.6 output is push-pull.
5	B5	**Port 4 Bit 5 Output Mode.** 0: P4.5 output is open-drain. 1: P4.5 output is push-pull.
4	B4	**Port 4 Bit 4 Output Mode.** 0: P4.4 output is open-drain. 1: P4.4 output is push-pull.
3	B3	**Port 4 Bit 3 Output Mode.** 0: P4.3 output is open-drain. 1: P4.3 output is push-pull.
2	B2	**Port 4 Bit 2 Output Mode.** 0: P4.2 output is open-drain. 1: P4.2 output is push-pull.
1	B1	**Port 4 Bit 1 Output Mode.** 0: P4.1 output is open-drain. 1: P4.1 output is push-pull.
0	B0	**Port 4 Bit 0 Output Mode.** 0: P4.0 output is open-drain. 1: P4.0 output is push-pull.

28.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 28.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

The 3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 28.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

28.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

- The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
- The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
- The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN bit in SPI0CFG and SPIEN bit in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
- The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

28.5. Serial Clock Phase and Polarity

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 28.5. For slave mode, the clock and data relationships are shown in Figure 28.6 and Figure 28.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

29.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

The interface exits Slave Receiver Mode after receiving a STOP. The interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 29.7 shows a typical slave write sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs *before* the ACK with hardware ACK generation disabled, and *after* the ACK when hardware ACK generation is enabled.

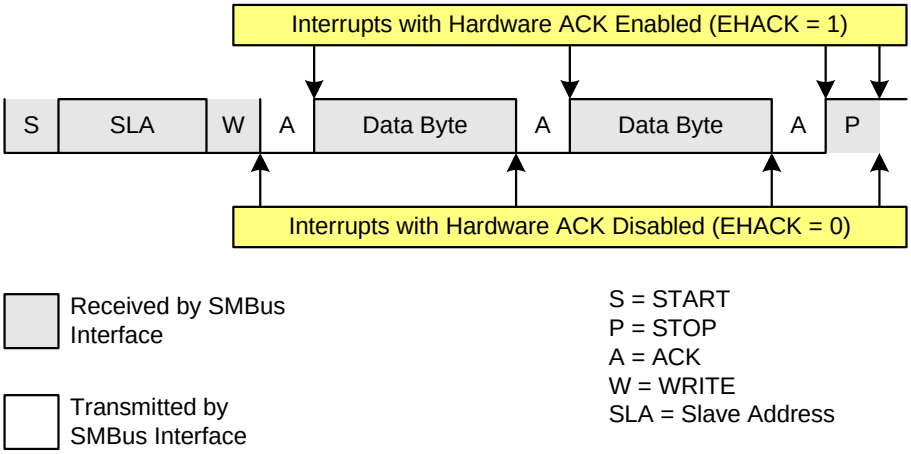

Figure 29.7. Typical Slave Write Sequence

Register 34.4. C2FPCTL: C2 Flash Programming Control

Bit	7	6	5	4	3	2	1	0
Name	C2FPCTL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
C2 Address: 0x02								

Table 34.4. C2FPCTL Register Bit Descriptions

Bit	Name	Function
7:0	C2FPCTL	**Flash Programming Control Register.** This register is used to enable flash programming via the C2 interface. To enable C2 flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 flash programming is enabled, a system reset must be issued to resume normal operation.