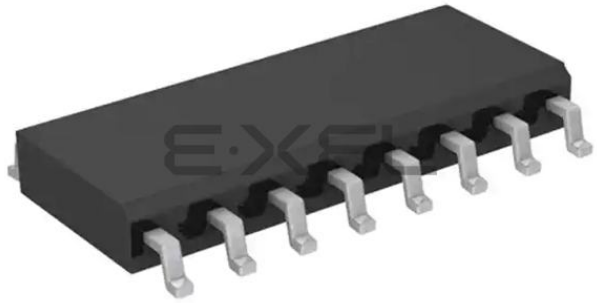


**Welcome to E-XFL.COM**

## 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 | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 13 |
| Program Memory Size | 8KB (8K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.2V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 16-SOIC (0.154", 3.90mm Width) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f863-c-gs |

- Byte-level bit reversal.
- Automatic CRC of flash contents on one or more 256-byte blocks.
- Initial seed selection of 0x0000 or 0xFFFF.

## 2.7. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- The core halts program execution.
- Module registers are initialized to their defined reset values unless the bits reset only with a power-on reset.
- External port pins are forced to a known state.
- Interrupts and timers are disabled.

All registers are reset to the predefined values noted in the register descriptions unless the bits only reset with a power-on reset. The contents of RAM are unaffected during a reset; any previously stored data is preserved as long as power is not lost.

The Port I/O latches are reset to 1 in open-drain mode. Weak pullups are enabled during and after the reset. For VDD Supply Monitor and power-on resets, the $\overline{\text{RST}}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal low-power oscillator. The Watchdog Timer is enabled with the Low Frequency Oscillator (LFO0) as its clock source. Program execution begins at location 0x0000.

## 2.8. On-Chip Debugging

The C8051F85x/86x 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.

**Table 3.2. Pin Definitions for C8051F850/1/2/3/4/5-GM and C8051F850/1/2/3/4/5-IM**

| Pin Name | Type | Pin Numbers | Crossbar Capability | Additional Digital Functions | Analog Functions |
|---|---|---|---|---|---|
| P1.0 | Standard I/O | 14 | Yes | P1MAT.0 | ADC0.8<br>CP1P.0<br>CP1N.0 |
| P1.1 | Standard I/O | 13 | Yes | P1MAT.1 | ADC0.9<br>CP1P.1<br>CP1N.1 |
| P1.2 | Standard I/O | 11 | Yes | P1MAT.2 | ADC0.10<br>CP1P.2<br>CP1N.2 |
| P1.3 | Standard I/O | 10 | Yes | P1MAT.3 | ADC0.11<br>CP1P.3<br>CP1N.3 |
| P1.4 | Standard I/O | 9 | Yes | P1MAT.4 | ADC0.12<br>CP1P.4<br>CP1N.4 |
| P1.5 | Standard I/O | 8 | Yes | P1MAT.5 | ADC0.13<br>CP1P.5<br>CP1N.5 |
| P1.6 | Standard I/O | 7 | Yes | P1MAT.6 | ADC0.14<br>CP1P.6<br>CP1N.6 |
| P2.0 /<br>C2D | Standard I/O /<br>C2 Debug Data | 6 | | | |

# 8. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051F85x/86x device family is shown in Figure 8.1.

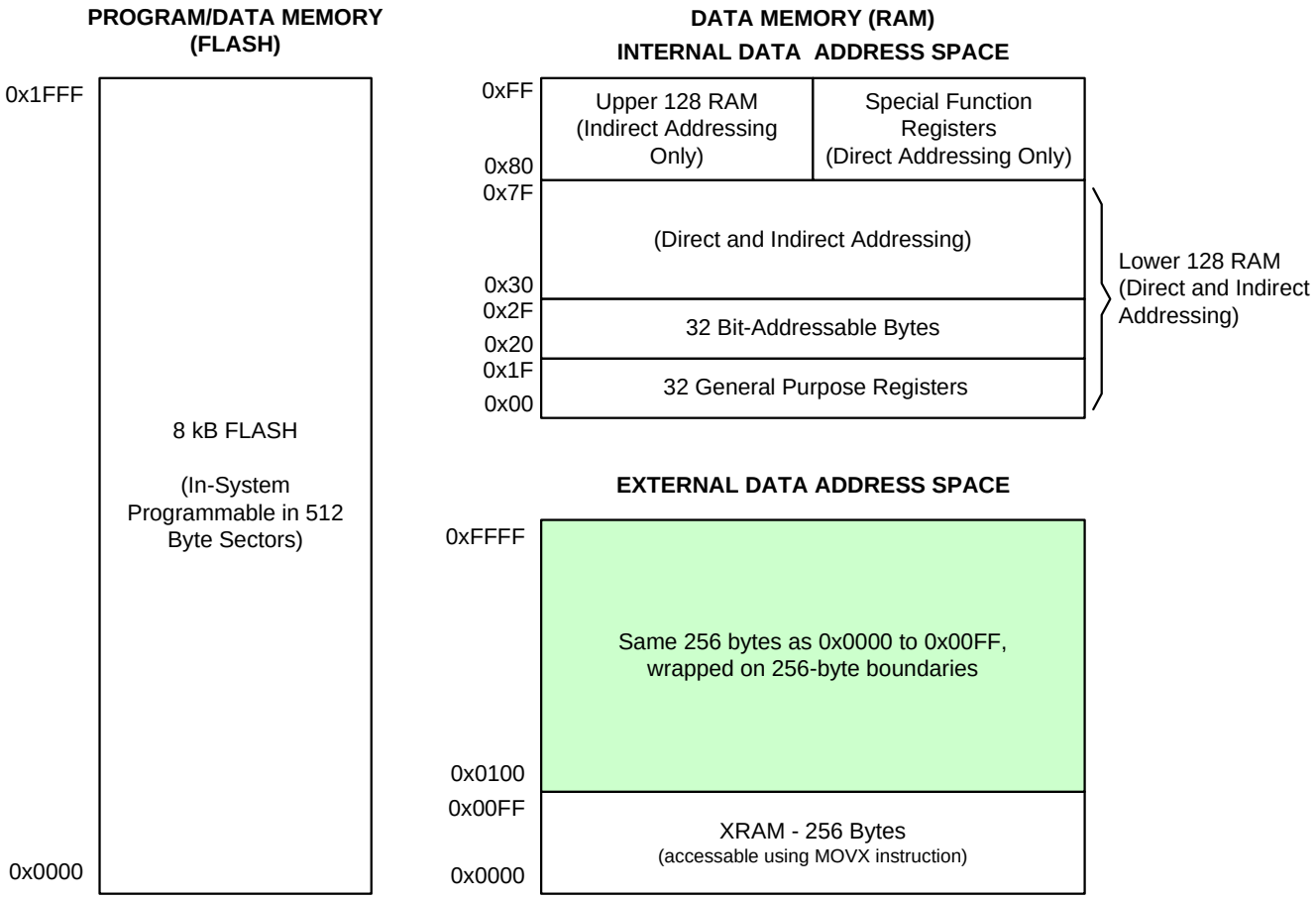



**Figure 8.1. C8051F85x/86x Memory Map (8 kB flash version shown)**

# 10. Flash Memory

On-chip, re-programmable flash memory is included for program code and non-volatile data storage. The flash memory is organized in 512-byte pages. It can be erased and written through the C2 interface or from firmware by overloading the MOVX instruction. Any individual byte in flash memory must only be written once between page erase operations.

## 10.1. Security Options

The CIP-51 provides security options to protect the flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the flash memory from accidental modification by software. PSWE must be explicitly set to '1' before software can modify the flash memory; both PSWE and PSEE must be set to '1' before software can erase flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located in flash user space offers protection of the flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. See Section "8. Memory Organization" on page 52 for the location of the security byte. The flash security mechanism allows the user to lock *n* 512-byte flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where *n* is the 1's complement number represented by the Security Lock Byte. **Note that the page containing the flash Security Lock Byte is unlocked when no other flash pages are locked (all bits of the Lock Byte are '1') and locked when any other flash pages are locked (any bit of the Lock Byte is '0').** An example is shown in Figure 10.1.

| | |
|---|---|
| Security Lock Byte: | 11111101b |
| 1s Complement: | 00000010b |
| Flash pages locked: | 3 (First two flash pages + Lock Byte Page) |

**Figure 10.1. Security Byte Decoding**

The level of flash security depends on the flash access method. The three flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 10.1 summarizes the flash security features of the C8051F85x/86x devices.

**Table 10.1. 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 | Flash Error Reset | Permitted |
| Read or Write page containing Lock Byte (if no pages are locked) | Permitted | Permitted | N/A |
| Read or Write page containing Lock Byte (if any page is locked) | Not Permitted | Flash Error Reset | Permitted |

## 12.2. Interrupt Control Registers

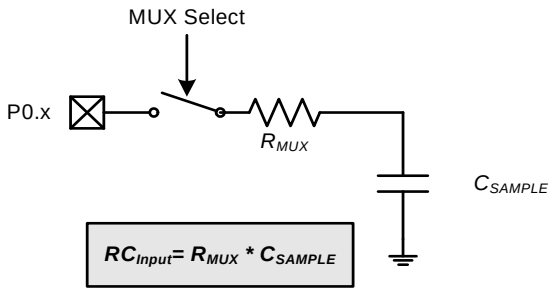
### Register 12.1. IE: Interrupt Enable

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | EA | ESPI0 | ET2 | ES0 | ET1 | EX1 | ET0 | EX0 |
| Type | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SFR Address: 0xA8 (bit-addressable)** | | | | | | | | |

**Table 12.2. IE Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7 | EA | **Enable All Interrupts.**<br>Globally enables/disables all interrupts and overrides individual interrupt mask settings.<br>0: Disable all interrupt sources.<br>1: Enable each interrupt according to its individual mask setting. |
| 6 | ESPI0 | **Enable SPI0 Interrupt.**<br>This bit sets the masking of the SPI0 interrupts.<br>0: Disable all SPI0 interrupts.<br>1: Enable interrupt requests generated by SPI0. |
| 5 | ET2 | **Enable Timer 2 Interrupt.**<br>This bit sets the masking of the Timer 2 interrupt.<br>0: Disable Timer 2 interrupt.<br>1: Enable interrupt requests generated by the TF2L or TF2H flags. |
| 4 | ES0 | **Enable UART0 Interrupt.**<br>This bit sets the masking of the UART0 interrupt.<br>0: Disable UART0 interrupt.<br>1: Enable UART0 interrupt. |
| 3 | ET1 | **Enable Timer 1 Interrupt.**<br>This bit sets the masking of the Timer 1 interrupt.<br>0: Disable all Timer 1 interrupt.<br>1: Enable interrupt requests generated by the TF1 flag. |
| 2 | EX1 | **Enable External Interrupt 1.**<br>This bit sets the masking of External Interrupt 1.<br>0: Disable external interrupt 1.<br>1: Enable interrupt requests generated by the INT1 input. |
| 1 | ET0 | **Enable Timer 0 Interrupt.**<br>This bit sets the masking of the Timer 0 interrupt.<br>0: Disable all Timer 0 interrupt.<br>1: Enable interrupt requests generated by the TF0 flag. |

*n* is the ADC resolution in bits (8/10/12).




**Note:** The value of CSAMPLE depends on the PGA Gain. See electrical specifications for details.

**Figure 14.4. ADC0 Equivalent Input Circuits**

### 14.2.5. Gain Setting

The ADC has gain settings of 1x and 0.5x. In 1x mode, the full scale reading of the ADC is determined directly by VREF. In 0.5x mode, the full-scale reading of the ADC occurs when the input voltage is VREF x 2. The 0.5x gain setting can be useful to obtain a higher input voltage range when using a small VREF voltage, or to measure input voltages that are between VREF and VDD. Gain settings for the ADC are controlled by the ADGN bit in register ADC0CF. Note that even with a gain setting of 0.5, voltages above the supply rail cannot be measured directly by the ADC.

## 14.3. 8-Bit Mode

Setting the ADC08BE bit in register ADC0CF to 1 will put the ADC in 8-bit mode. In 8-bit mode, only the 8 MSBs of data are converted, allowing the conversion to be completed in fewer SAR clock cycles than a 10-bit conversion. The two LSBs of a conversion are always 00 in this mode, and the ADC0L register will always read back 0x00.

## 14.4. 12-Bit Mode

When configured for 12-bit conversions, the ADC performs four 10-bit conversions using four different reference voltages and combines the results into a single 12-bit value. Unlike simple averaging techniques, this method provides true 12-bit resolution of AC or DC input signals without depending on noise to provide dithering. The converter also employs a hardware dynamic element matching algorithm that reconfigures the largest elements of the internal DAC for each of the four 10-bit conversions. This reconfiguration cancels any matching errors and enables the converter to achieve 12-bit linearity performance to go along with its 12-bit resolution.

The 12-bit mode is enabled by setting the AD12BE bit in register ADC0AC to logic 1 and configuring the ADC in burst mode (ADBMEN = 1) for four or more conversions. The conversion can be initiated using any of the conversion start sources, and the 12-bit result will appear in the ADC0H and ADC0L registers. Since the 12-bit result is formed from a combination of four 10-bit results, the maximum output value is 4 x (1023) = 4092, rather than the max value of (2^12 – 1) = 4095 that is produced by a traditional 12-bit converter. To further increase resolution, the burst mode repeat value may be configured to any multiple of four conversions. For example, if a repeat value of 16 is selected, the ADC0 output will be a 14-bit number (sum of four 12-bit numbers) with 13 effective bits of resolution.

The AD12SM bit in register ADC0TK controls when the ADC will track and sample the input signal. When AD12SM is set to 1, the selected input signal will be tracked before the first conversion of a set and held internally during all four conversions. When AD12SM is cleared to 0, the ADC will track and sample the selected input before each of the four conversions in a set. When maximum throughput (180-200 ksps) is

## Register 14.10. ADC0GTL: ADC0 Greater-Than Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ADC0GTL | | | | | | | |
| Type | RW | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **SFR Address: 0xC3** | | | | | | | | |

**Table 14.13. ADC0GTL Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7:0 | ADC0GTL | **Greater-Than Low Byte.**<br>Least Significant Byte of the 16-bit Greater-Than window compare register. |
| **Note:** In 8-bit mode, this register should be set to 0x00. | | |

## Register 14.11. ADC0LTH: ADC0 Less-Than High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ADC0LTH | | | | | | | |
| Type | RW | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SFR Address: 0xC6** | | | | | | | | |

**Table 14.14. ADC0LTH Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7:0 | ADC0LTH | **Less-Than High Byte.**<br>Most Significant Byte of the 16-bit Less-Than window compare register. |

## Register 15.6. PSW: Program Status Word

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CY | AC | F0 | RS | | OV | F1 | PARITY |
| Type | RW | RW | RW | RW | | RW | RW | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SFR Address: 0xD0 (bit-addressable)** | | | | | | | | |

**Table 15.7. PSW Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7 | CY | **Carry Flag.**<br>This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations. |
| 6 | AC | **Auxiliary Carry Flag.**<br>This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations. |
| 5 | F0 | **User Flag 0.**<br>This is a bit-addressable, general purpose flag for use under software control. |
| 4:3 | RS | **Register Bank Select.**<br>These bits select which register bank is used during register accesses.<br>00: Bank 0, Addresses 0x00-0x07<br>01: Bank 1, Addresses 0x08-0x0F<br>10: Bank 2, Addresses 0x10-0x17<br>11: Bank 3, Addresses 0x18-0x1F |
| 2 | OV | **Overflow Flag.**<br>This bit is set to 1 under the following circumstances:<br>1. An ADD, ADDC, or SUBB instruction causes a sign-change overflow.<br>2. A MUL instruction results in an overflow (result is greater than 255).<br>3. A DIV instruction causes a divide-by-zero condition.<br>The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases. |
| 1 | F1 | **User Flag 1.**<br>This is a bit-addressable, general purpose flag for use under software control. |
| 0 | PARITY | **Parity Flag.**<br>This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even. |

## Register 20.4. PCA0CLR: PCA Comparator Clear Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CPCPOL | Reserved | | | | CPCE2 | CPCE1 | CPCE0 |
| Type | RW | R | | | | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SFR Address: 0x9C** | | | | | | | | |

**Table 20.6. PCA0CLR Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7 | CPCPOL | **Comparator Clear Polarity.**<br>Selects the polarity of the comparator result that will clear the PCA channel(s).<br>0: PCA channel(s) will be cleared when comparator result goes logic low.<br>1: PCA channel(s) will be cleared when comparator result goes logic high. |
| 6:3 | Reserved | Must write reset value. |
| 2 | CPCE2 | **Comparator Clear Enable for CEX2.**<br>Enables the comparator clear function on PCA channel 2. |
| 1 | CPCE1 | **Comparator Clear Enable for CEX1.**<br>Enables the comparator clear function on PCA channel 1. |
| 0 | CPCE0 | **Comparator Clear Enable for CEX0.**<br>Enables the comparator clear function on PCA channel 0. |

## Register 20.11. PCA0CENT: PCA Center Alignment Enable

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | CEX2CEN | CEX1CEN | CEX0CEN |
| Type | R | | | | | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SFR Address: 0x9E** | | | | | | | | |

**Table 20.13. PCA0CENT Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7:3 | Reserved | Must write reset value. |
| 2 | CEX2CEN | **CEX2 Center Alignment Enable.**<br>Selects the alignment properties of the CEX2 output channel when operated in any of the PWM modes. This bit does not affect the operation of non-PWM modes.<br>0: Edge-aligned.<br>1: Center-aligned. |
| 1 | CEX1CEN | **CEX1 Center Alignment Enable.**<br>Selects the alignment properties of the CEX1 output channel when operated in any of the PWM modes. This bit does not affect the operation of non-PWM modes.<br>0: Edge-aligned.<br>1: Center-aligned. |
| 0 | CEX0CEN | **CEX0 Center Alignment Enable.**<br>Selects the alignment properties of the CEX0 output channel when operated in any of the PWM modes. This bit does not affect the operation of non-PWM modes.<br>0: Edge-aligned.<br>1: Center-aligned. |

## Register 21.6. P0MAT: Port 0 Match

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | P0MAT | | | | | | | |
| Type | RW | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **SFR Address: 0xFD** | | | | | | | | |

**Table 21.9. P0MAT Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7:0 | P0MAT | **Port 0 Match Value.**<br>Match comparison value used on P0 pins for bits in P0MASK which are set to 1.<br>0: P0.x pin logic value is compared with logic LOW.<br>1: P0.x pin logic value is compared with logic HIGH. |

## 23.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 23.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

### 23.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

### 23.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

### 23.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 23.2, Figure 23.3, and Figure 23.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device.

## 23.2. 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 23.2 shows a connection diagram between two master devices and a single slave 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 23.3 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 23.4 shows a connection diagram for a master device and a slave device in 4-wire mode.

## 23.7. SPI Control Registers

### Register 23.1. SPI0CFG: SPI0 Configuration

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SPIBSY | MSTEN | CKPHA | CKPOL | SLVSEL | NSSIN | SRMT | RXBMT |
| Type | R | RW | RW | RW | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| **SFR Address: 0xA1** | | | | | | | | |

**Table 23.2. SPI0CFG Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7 | SPIBSY | **SPI Busy.**<br>This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode). |
| 6 | MSTEN | **Master Mode Enable.**<br>0: Disable master mode. Operate in slave mode.<br>1: Enable master mode. Operate as a master. |
| 5 | CKPHA | **SPI0 Clock Phase.**<br>0: Data centered on first edge of SCK period.<br>1: Data centered on second edge of SCK period. |
| 4 | CKPOL | **SPI0 Clock Polarity.**<br>0: SCK line low in idle state.<br>1: SCK line high in idle state. |
| 3 | SLVSEL | **Slave Selected Flag.**<br>This bit is 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.**<br>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).**<br>This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when 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. | | |

**Table 24.7. SMB0CF Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 1:0 | SMBCS | **SMBus0 Clock Source Selection.**<br>These two bits select the SMBus0 clock source, which is used to generate the SMBus0 bit rate. See the SMBus clock timing section for additional details.<br>00: Timer 0 Overflow<br>01: Timer 1 Overflow<br>10: Timer 2 High Byte Overflow<br>11: Timer 2 Low Byte Overflow |

## Register 25.17. TMR3H: Timer 3 High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3H | | | | | | | |
| Type | RW | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SFR Address: 0x95** | | | | | | | | |

**Table 25.19. TMR3H Register Bit Descriptions**

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3H | **Timer 3 High Byte.**<br>In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value. |