

Welcome to [E-XFL.COM](https://www.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	I <sup>2</sup> C, 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	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f863-c-gsr">https://www.e-xfl.com/product-detail/silicon-labs/c8051f863-c-gsr</a>

# C8051F85x-86x

---

24.2. SMBus Configuration.....	234
24.3. SMBus Operation .....	234
24.3.1. Transmitter vs. Receiver .....	235
24.3.2. Arbitration.....	235
24.3.3. Clock Low Extension.....	235
24.3.4. SCL Low Timeout.....	235
24.3.5. SCL High (SMBus Free) Timeout .....	236
24.4. Using the SMBus.....	236
24.4.1. SMBus Configuration Register.....	236
24.4.2. SMBus Pin Swap .....	238
24.4.3. SMBus Timing Control .....	238
24.4.4. SMB0CN Control Register .....	238
24.4.5. Hardware Slave Address Recognition .....	240
24.4.6. Data Register .....	241
24.5. SMBus Transfer Modes.....	242
24.5.1. Write Sequence (Master) .....	242
24.5.2. Read Sequence (Master) .....	243
24.5.3. Write Sequence (Slave) .....	244
24.5.4. Read Sequence (Slave).....	245
24.6. SMBus Status Decoding.....	245
24.7. I2C / SMBus Control Registers.....	251
<b>25. Timers (Timer0, Timer1, Timer2 and Timer3) .....</b>	<b>259</b>
25.1. Timer 0 and Timer 1 .....	261
25.1.1. Mode 0: 13-bit Counter/Timer .....	262
25.1.2. Mode 1: 16-bit Counter/Timer .....	263
25.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload.....	264
25.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only).....	265
25.2. Timer 2 and Timer 3 .....	266
25.2.1. 16-bit Timer with Auto-Reload.....	266
25.2.2. 8-bit Timers with Auto-Reload.....	267
25.2.3. Capture Mode .....	268
25.3. Timer Control Registers.....	269
<b>26. Universal Asynchronous Receiver/Transmitter (UART0) .....</b>	<b>289</b>
26.1. Enhanced Baud Rate Generation.....	289
26.2. Operational Modes .....	291
26.2.1. 8-Bit UART .....	291
26.2.2. 9-Bit UART .....	292
26.3. Multiprocessor Communications .....	293
26.4. UART Control Registers.....	295
<b>27. Watchdog Timer (WDT0) .....</b>	<b>298</b>
27.1. Enabling / Resetting the WDT .....	299
27.2. Disabling the WDT.....	299
27.3. Disabling the WDT Lockout.....	299
27.4. Setting the WDT Interval .....	299
27.5. Watchdog Timer Control Registers .....	300

---

---

### 2.1.3.1. Normal Mode

Normal mode encompasses the typical full-speed operation. The power consumption of the device in this mode will vary depending on the system clock speed and any analog peripherals that are enabled.

### 2.1.3.2. Idle Mode

Setting the IDLE bit in PCON causes the hardware to halt the CPU and enter idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the IDLE bit to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

### 2.1.3.3. Stop Mode (Regulator On)

Setting the STOP bit in PCON when STOPCF in REG0CN is clear causes the controller core to enter stop mode as soon as the instruction that sets the bit completes execution. In stop mode the internal oscillator, CPU, and all digital peripherals are stopped. Each analog peripheral may be shut down individually prior to entering stop mode. Stop mode can only be terminated by an internal or external reset.

### 2.1.3.4. Shutdown Mode (Regulator Off)

Shutdown mode is an extension of the normal stop mode operation. Setting the STOP bit in PCON when STOPCF in REG0CN is also set causes the controller core to enter shutdown mode as soon as the instruction that sets the bit completes execution, and then the internal regulator is powered down. In shutdown mode, all core functions, memories and peripherals are powered off. An external pin reset or power-on reset is required to exit shutdown mode.

## 2.2. I/O

### 2.2.1. General Features

The C8051F85x/86x ports have the following features:

- Push-pull or open-drain output modes and analog or digital modes.
- Port Match allows the device to recognize a change on a port pin value and wake from idle mode or generate an interrupt.
- Internal pull-up resistors can be globally enabled or disabled.
- Two external interrupts provide unique interrupt vectors for monitoring time-critical events.
- Above-rail tolerance allows 5 V interface when device is powered.

### 2.2.2. Crossbar

The C8051F85x/86x devices have a digital peripheral crossbar with the following features:

- Flexible peripheral assignment to port pins.
- Pins can be individually skipped to move peripherals as needed for design or layout considerations.

The crossbar has a fixed priority for each I/O function and assigns these functions to the port pins. When a digital resource is selected, the least-significant unassigned port pin is assigned to that resource. If a port pin is assigned, the crossbar skips that pin when assigning the next selected resource. Additionally, the crossbar will skip port pins whose associated bits in the PnSKIP registers are set. This provides some flexibility when designing a system: pins involved with sensitive analog measurements can be moved away from digital I/O and peripherals can be moved around the chip as needed to ease layout constraints.

---

**Table 3.1. Pin Definitions for C8051F850/1/2/3/4/5-GU and C8051F850/1/2/3/4/5-IU**

<b>Pin Name</b>	<b>Type</b>	<b>Pin Numbers</b>	<b>Crossbar Capability</b>	<b>Additional Digital Functions</b>	<b>Analog Functions</b>
N/C	No Connection	1 13 24			

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

<b>Pin Name</b>	<b>Type</b>	<b>Pin Numbers</b>	<b>Crossbar Capability</b>	<b>Additional Digital Functions</b>	<b>Analog Functions</b>
P0.0	Standard I/O	2	Yes	P0MAT.0 INT0.0 INT1.0	ADC0.0 CP0P.0 CP0N.0 VREF
P0.1	Standard I/O	1	Yes	P0MAT.1 INT0.1 INT1.1	ADC0.1 CP0P.1 CP0N.1 AGND
P0.2	Standard I/O	20	Yes	P0MAT.2 INT0.2 INT1.2	ADC0.2 CP0P.2 CP0N.2
P0.3	Standard I/O	19	Yes	P0MAT.3 EXTCLK INT0.3 INT1.3	ADC0.3 CP0P.3 CP0N.3
P0.4	Standard I/O	18	Yes	P0MAT.4 INT0.4 INT1.4	ADC0.4 CP0P.4 CP0N.4
P0.5	Standard I/O	17	Yes	P0MAT.5 INT0.5 INT1.5	ADC0.5 CP0P.5 CP0N.5
P0.6	Standard I/O	16	Yes	P0MAT.6 CNVSTR INT0.6 INT1.6	ADC0.6 CP0P.6 CP0N.6
P0.7	Standard I/O	15	Yes	P0MAT.7 INT0.7 INT1.7	ADC0.7 CP0P.7 CP0N.7

---

**Register 12.2. IP: Interrupt Priority**

---

Bit	7	6	5	4	3	2	1	0
Name	Reserved	PSPiO	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	1	0	0	0	0	0	0	0
<b>SFR Address: 0xB8 (bit-addressable)</b>								

**Table 12.3. IP Register Bit Descriptions**

Bit	Name	Function
7	Reserved	Must write reset value.
6	PSPiO	<b>Serial Peripheral Interface (SPI0) Interrupt Priority Control.</b> This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	<b>Timer 2 Interrupt Priority Control.</b> This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	<b>UART0 Interrupt Priority Control.</b> This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	<b>Timer 1 Interrupt Priority Control.</b> This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	<b>External Interrupt 1 Priority Control.</b> This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	<b>Timer 0 Interrupt Priority Control.</b> This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	<b>External Interrupt 0 Priority Control.</b> This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

---

**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	<b>Greater-Than Low Byte.</b> Least Significant Byte of the 16-bit Greater-Than window compare register.
<b>Note:</b> In 8-bit mode, this register should be set to 0x00.		

---

## 15.4. CPU Core Registers

---

### Register 15.1. DPL: Data Pointer Low

---

Bit	7	6	5	4	3	2	1	0
Name	DPL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x82								

**Table 15.2. DPL Register Bit Descriptions**

Bit	Name	Function
7:0	DPL	<b>Data Pointer Low.</b> The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed flash memory or XRAM.



---



---

**Register 17.6. CPT1MX: Comparator 1 Multiplexer Selection**


---

Bit	7	6	5	4	3	2	1	0
Name	CMXN				CMXP			
Type	RW				RW			
Reset	1	1	1	1	1	1	1	1

**SFR Address: 0xAA**

**Table 17.10. CPT1MX Register Bit Descriptions**

Bit	Name	Function
7:4	CMXN	<b>Comparator 1 Negative Input MUX Selection.</b> 0000: External pin CP1N.0 0001: External pin CP1N.1 0010: External pin CP1N.2 0011: External pin CP1N.3 0100: External pin CP1N.4 0101: External pin CP1N.5 0110: External pin CP1N.6 0111: External pin CP1N.7 1000: GND 1001-1111: Reserved.
3:0	CMXP	<b>Comparator 1 Positive Input MUX Selection.</b> 0000: External pin CP1P.0 0001: External pin CP1P.1 0010: External pin CP1P.2 0011: External pin CP1P.3 0100: External pin CP1P.4 0101: External pin CP1P.5 0110: External pin CP1P.6 0111: External pin CP1P.7 1000: Internal LDO output 1001-1111: Reserved.

---

disable the comparison, and prevent the match edge from occurring. Note that although the PCA0CPn compare register determines the duty cycle, it is not always appropriate for firmware to update this register directly. See the sections on 8 to 11-bit and 16-bit PWM mode for additional details on adjusting duty cycle in the various modes.

$$\text{Duty Cycle} = \frac{(2^N - \text{PCA0CPn})}{2^N}$$

**Equation 20.2. N-bit Edge-Aligned PWM Duty Cycle With CEXnPOL = 0 (N = PWM resolution)**

$$\text{Duty Cycle} = \frac{\text{PCA0CPn}}{2^N}$$

**Equation 20.3. N-bit Edge-Aligned PWM Duty Cycle With CEXnPOL = 0 (N = PWM resolution)**

---

#### 20.4.4. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8 through 11-bit) PWM modes. The entire PCA0CP register is used to determine the duty cycle in 16-bit PWM mode.

To output a varying duty cycle, new value writes should be synchronized with the PCA CCFn match flag to ensure seamless updates.

16-Bit PWM mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, the match interrupt flag should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a match edge or up edge occurs. The CF flag in PCA0CN can be used to detect the overflow or down edge.

**Important Note About Capture/Compare Registers:** When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

---

**Register 20.5. PCA0CPM0: PCA Capture/Compare Mode**

---

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

**SFR Address: 0xDA**

**Table 20.7. PCA0CPM0 Register Bit Descriptions**

Bit	Name	Function
7	PWM16	<b>16-bit Pulse Width Modulation Enable.</b> This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	<b>Comparator Function Enable.</b> This bit enables the comparator function.
5	CAPP	<b>Capture Positive Function Enable.</b> This bit enables the positive edge capture capability.
4	CAPN	<b>Capture Negative Function Enable.</b> This bit enables the negative edge capture capability.
3	MAT	<b>Match Function Enable.</b> This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF0 bit in the PCA0MD register to be set to logic 1.
2	TOG	<b>Toggle Function Enable.</b> This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX0 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	<b>Pulse Width Modulation Mode Enable.</b> This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX0 pin. 8 to 11-bit PWM is used if PWM16 is cleared; 16-bit mode is used if PWM16 is set to logic 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	<b>Capture/Compare Flag Interrupt Enable.</b> This bit sets the masking of the Capture/Compare Flag (CCF0) interrupt. 0: Disable CCF0 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF0 is set.

## 21.7. Port I/O and Pin Configuration Control Registers

### Register 21.1. XBR0: Port I/O Crossbar 0

Bit	7	6	5	4	3	2	1	0
Name	SYSCKE	CP1AE	CP1E	CP0AE	CP0E	SMB0E	SPI0E	URT0E
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xE1								

Table 21.4. XBR0 Register Bit Descriptions

Bit	Name	Function
7	SYSCKE	<b>SYSCCLK Output Enable.</b> 0: SYSCCLK unavailable at Port pin. 1: SYSCCLK output routed to Port pin.
6	CP1AE	<b>Comparator1 Asynchronous Output Enable.</b> 0: Asynchronous CP1 unavailable at Port pin. 1: Asynchronous CP1 routed to Port pin.
5	CP1E	<b>Comparator1 Output Enable.</b> 0: CP1 unavailable at Port pin. 1: CP1 routed to Port pin.
4	CP0AE	<b>Comparator0 Asynchronous Output Enable.</b> 0: Asynchronous CP0 unavailable at Port pin. 1: Asynchronous CP0 routed to Port pin.
3	CP0E	<b>Comparator0 Output Enable.</b> 0: CP0 unavailable at Port pin. 1: CP0 routed to Port pin.
2	SMB0E	<b>SMBus0 I/O Enable.</b> 0: SMBus0 I/O unavailable at Port pins. 1: SMBus0 I/O routed to Port pins.
1	SPI0E	<b>SPI I/O Enable.</b> 0: SPI I/O unavailable at Port pins. 1: SPI I/O routed to Port pins. The SPI can be assigned either 3 or 4 GPIO pins.
0	URT0E	<b>UART I/O Output Enable.</b> 0: UART I/O unavailable at Port pin. 1: UART TX, RX routed to Port pins P0.4 and P0.5.

---

**Register 21.2. XBR1: Port I/O Crossbar 1**

---

Bit	7	6	5	4	3	2	1	0
Name	Reserved		T2E	T1E	T0E	ECIE	PCA0ME	
Type	R	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0

**SFR Address: 0xE2****Table 21.5. XBR1 Register Bit Descriptions**

Bit	Name	Function
7:6	Reserved	Must write reset value.
5	T2E	<b>T2 Enable.</b> 0: T2 unavailable at Port pin. 1: T2 routed to Port pin.
4	T1E	<b>T1 Enable.</b> 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
3	T0E	<b>T0 Enable.</b> 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
2	ECIE	<b>PCA0 External Counter Input Enable.</b> 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
1:0	PCA0ME	<b>PCA Module I/O Enable Bits.</b> 00: All PCA I/O unavailable at Port pins. 01: CEX0 routed to Port pin. 10: CEX0, CEX1 routed to Port pins. 11: CEX0, CEX1, CEX2 routed to Port pins.

---

---

**Register 21.13. P1: Port 1 Pin Latch**

---

Bit	7	6	5	4	3	2	1	0
Name	P1							
Type	RW							
Reset	1	1	1	1	1	1	1	1
<b>SFR Address: 0x90 (bit-addressable)</b>								

**Table 21.16. P1 Register Bit Descriptions**

Bit	Name	Function
7:0	P1	<b>Port 1 Data.</b> Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.
<b>Note:</b> Port 1 consists of 8 bits (P1.0-P1.7) on QSOP24 packages and 7 bits (P1.0-P1.6) on QFN20 packages and 4 bits (P1.0-P1.3) on SOIC16 packages.		

---

### 23.3. 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 23.4 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 23.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

### 23.4. SPI0 Interrupt Sources

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

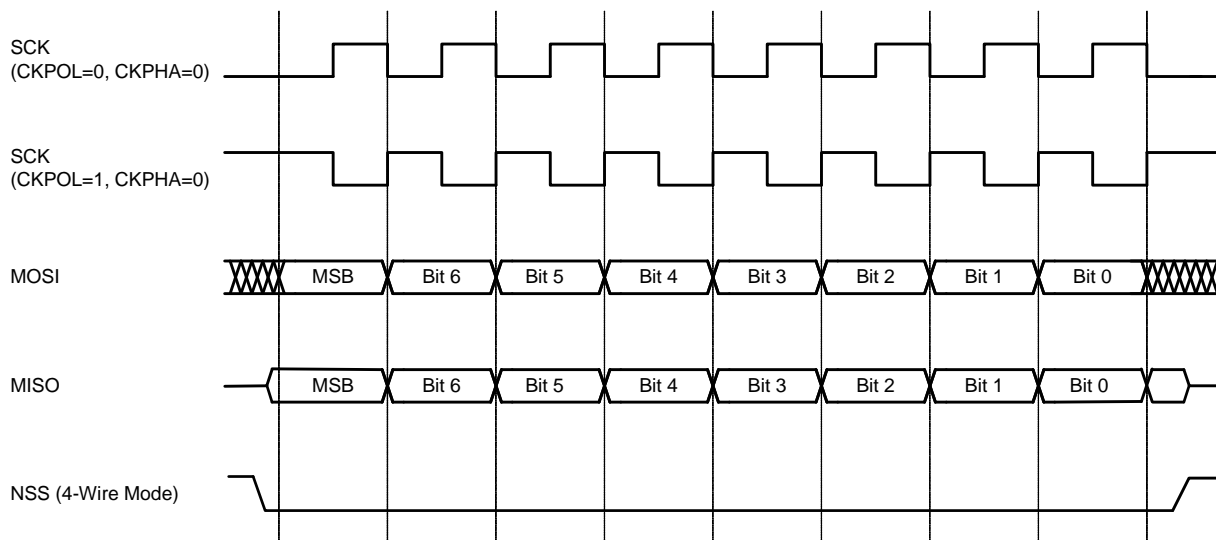
All of the following bits must be cleared by software.

- 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 and SPIEN bits 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.

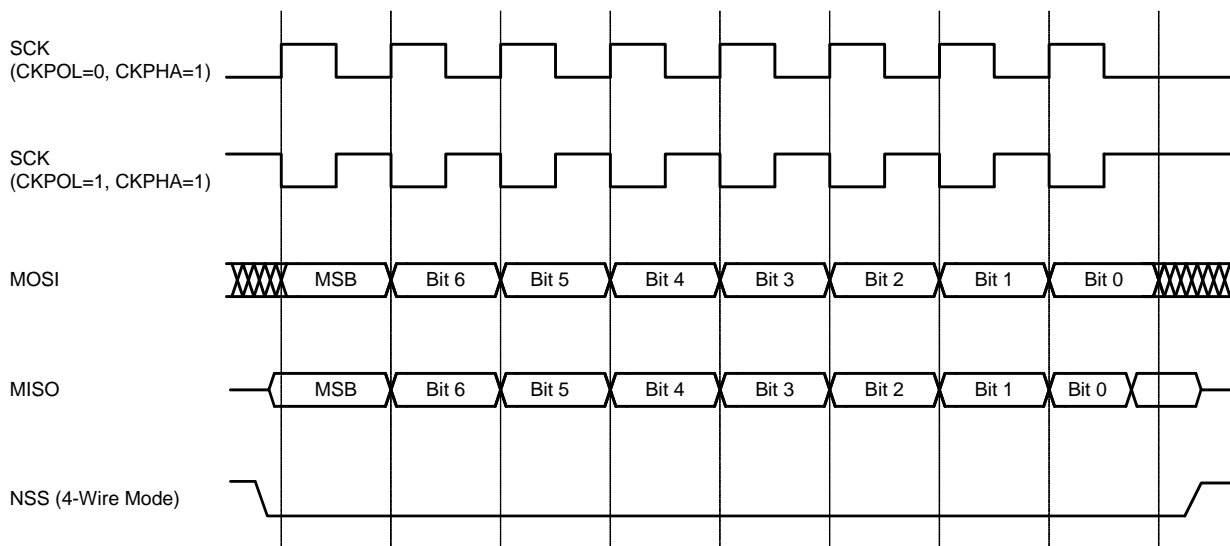
### 23.5. Serial Clock Phase and Polarity

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





**Figure 23.6. Slave Mode Data/Clock Timing (CKPHA = 0)**



**Figure 23.7. Slave Mode Data/Clock Timing (CKPHA = 1)**

## 23.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

---

**Table 24.4. Hardware Address Recognition Examples (EHACK = 1)**

Hardware Slave Address SLV	Slave Address Mask SLVM	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

#### 24.4.6. Data Register

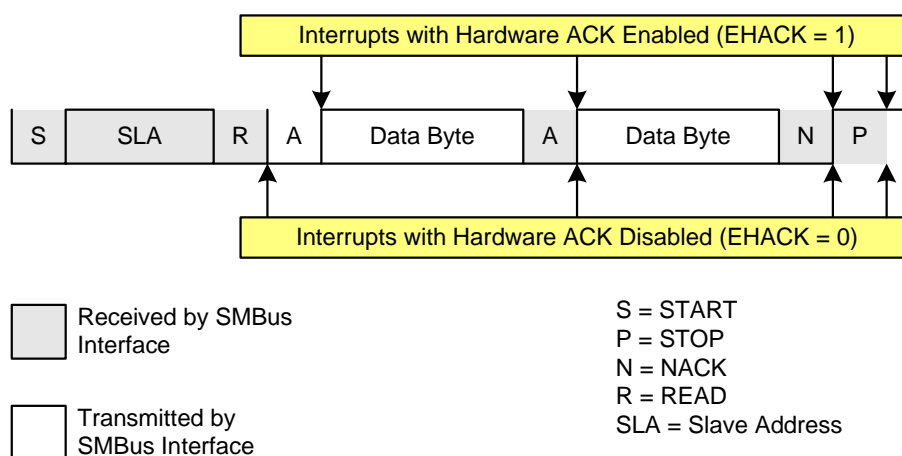
The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared 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.

#### 24.5.4. Read Sequence (Slave)

During a read sequence, an SMBus master reads data from a slave device. The slave in this transfer will be a receiver during the address byte, and a transmitter during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode (to receive the slave address) when a START followed by a slave address and direction bit (READ 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 transmitted. If the received slave address is acknowledged, data should be written to SMB0DAT to be transmitted. The interface enters slave transmitter mode, and transmits one or more bytes of data. After each byte is transmitted, the master sends an acknowledge bit; if the acknowledge bit is an ACK, SMB0DAT should be written with the next data byte. If the acknowledge bit is a NACK, SMB0DAT should not be written to before SI is cleared (an error condition may be generated if SMB0DAT is written following a received NACK while in slave transmitter mode). The interface exits slave transmitter mode after receiving a STOP. The interface will switch to slave receiver mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 24.8 shows a typical slave read sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.



**Figure 24.8. Typical Slave Read Sequence**

#### 24.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. The appropriate actions to take in response to an SMBus event depend on whether hardware slave address recognition and ACK generation is enabled or disabled. Table 24.5 describes the typical actions when hardware slave address recognition and ACK generation is disabled. Table 24.6 describes the typical actions when hardware slave address recognition and ACK generation is enabled. In the tables, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed by hardware but do not conform to the SMBus specification.

## 25.2. Timer 2 and Timer 3

Timer 2 and Timer 3 are functionally equivalent, with the only differences being the top-level connections to other parts of the system, as detailed in Table 25.1 and Table 25.2.

The timers are 16 bits wide, formed by two 8-bit SFRs: TMRnL (low byte) and TMRnH (high byte). Each timer may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The TnSPLIT bit in TMRnCN defines the timer operation mode.

The timers may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

### 25.2.1. 16-bit Timer with Auto-Reload

When TnSPLIT is zero, the timer operates as a 16-bit timer with auto-reload. In this mode, the timer may be configured to clock from SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the timer reload registers (TMRnRLH and TMRnRLL) is loaded into the main timer count register as shown in Figure 25.4, and the High Byte Overflow Flag (TFnH) is set. If the timer interrupts are enabled, an interrupt will be generated on each timer overflow. Additionally, if the timer interrupts are enabled and the TFnLEN bit is set, an interrupt will be generated each time the lower 8 bits (TMRnL) overflow from 0xFF to 0x00.

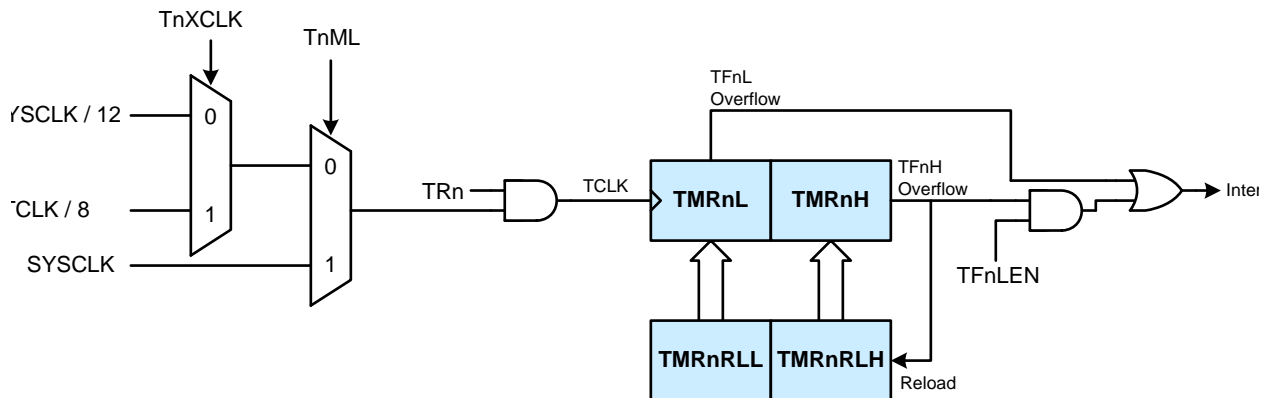


Figure 25.4. 16-Bit Mode Block Diagram

---

**Register 25.15. TMR3RLH: Timer 3 Reload High Byte**

---

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x93								

**Table 25.17. TMR3RLH Register Bit Descriptions**

Bit	Name	Function
7:0	TMR3RLH	<b>Timer 3 Reload High Byte.</b> When operating in one of the auto-reload modes, TMR3RLH holds the reload value for the high byte of Timer 3 (TMR3H). When operating in capture mode, TMR3RLH is the captured value of TMR3H.