



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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, WDT
Number of I/O	54
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f703-gq">https://www.e-xfl.com/product-detail/silicon-labs/c8051f703-gq</a>

# C8051F70x/71x

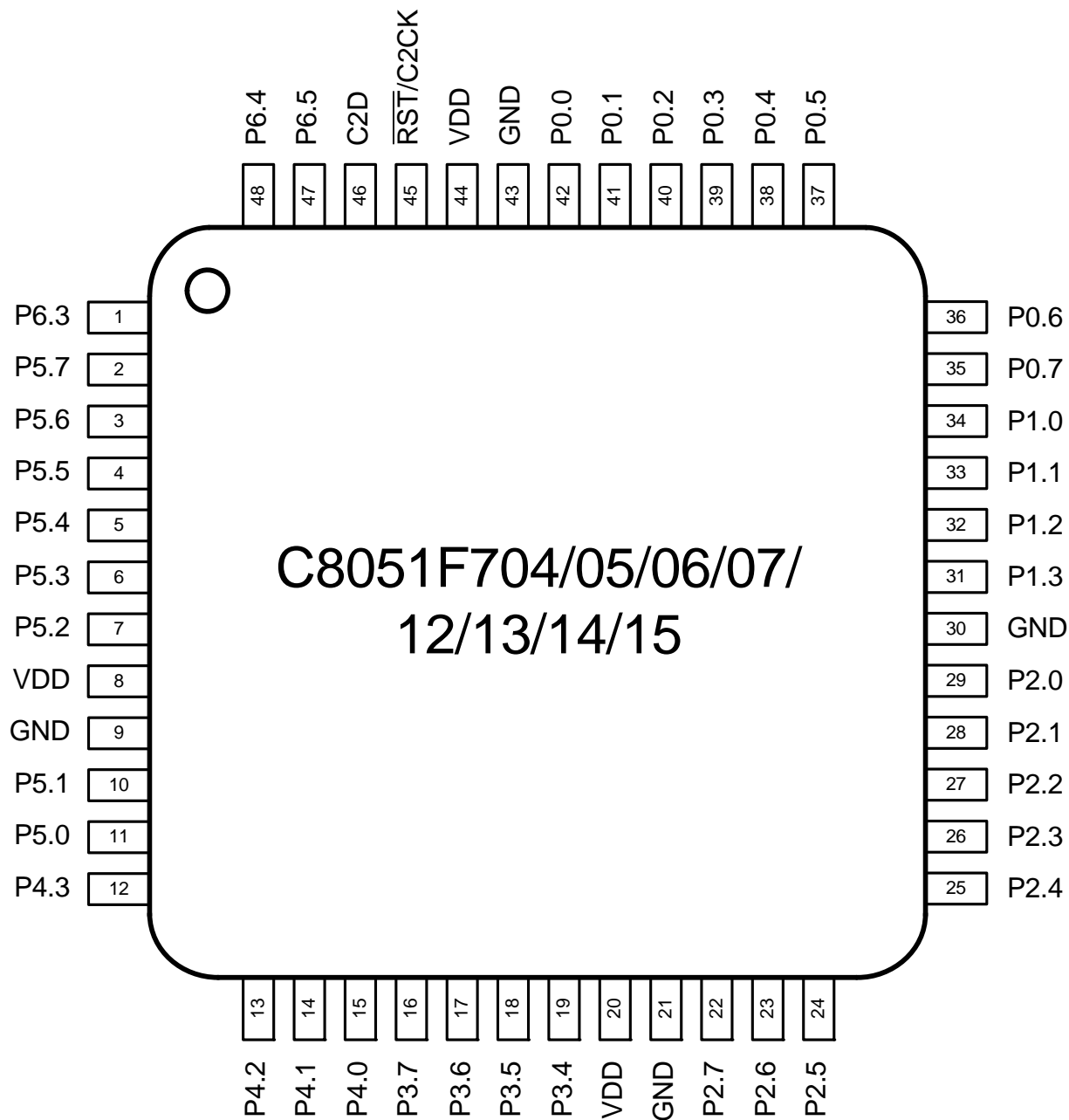


Figure 3.2. C8051F7xx-GQ QFP48 Pinout Diagram (Top View)

## 10.1. Output Code Formatting

The ADC measures the input voltage with reference to GND. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit. Conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

## 10.2. 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, and the ADC0H register holds the results. The AD0LJST bit is ignored for 8-bit mode. 8-bit conversions take two fewer SAR clock cycles than 10-bit conversions, so the conversion is completed faster, and a 500 kps sampling rate can be achieved with a slower SAR clock.

## 10.3. Modes of Operation

ADC0 has a maximum conversion speed of 500 kps. The ADC0 conversion clock is a divided version of the system clock, determined by the AD0SC bits in the ADC0CF register.

### 10.3.1. Starting a Conversion

A conversion can be initiated in one of six ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

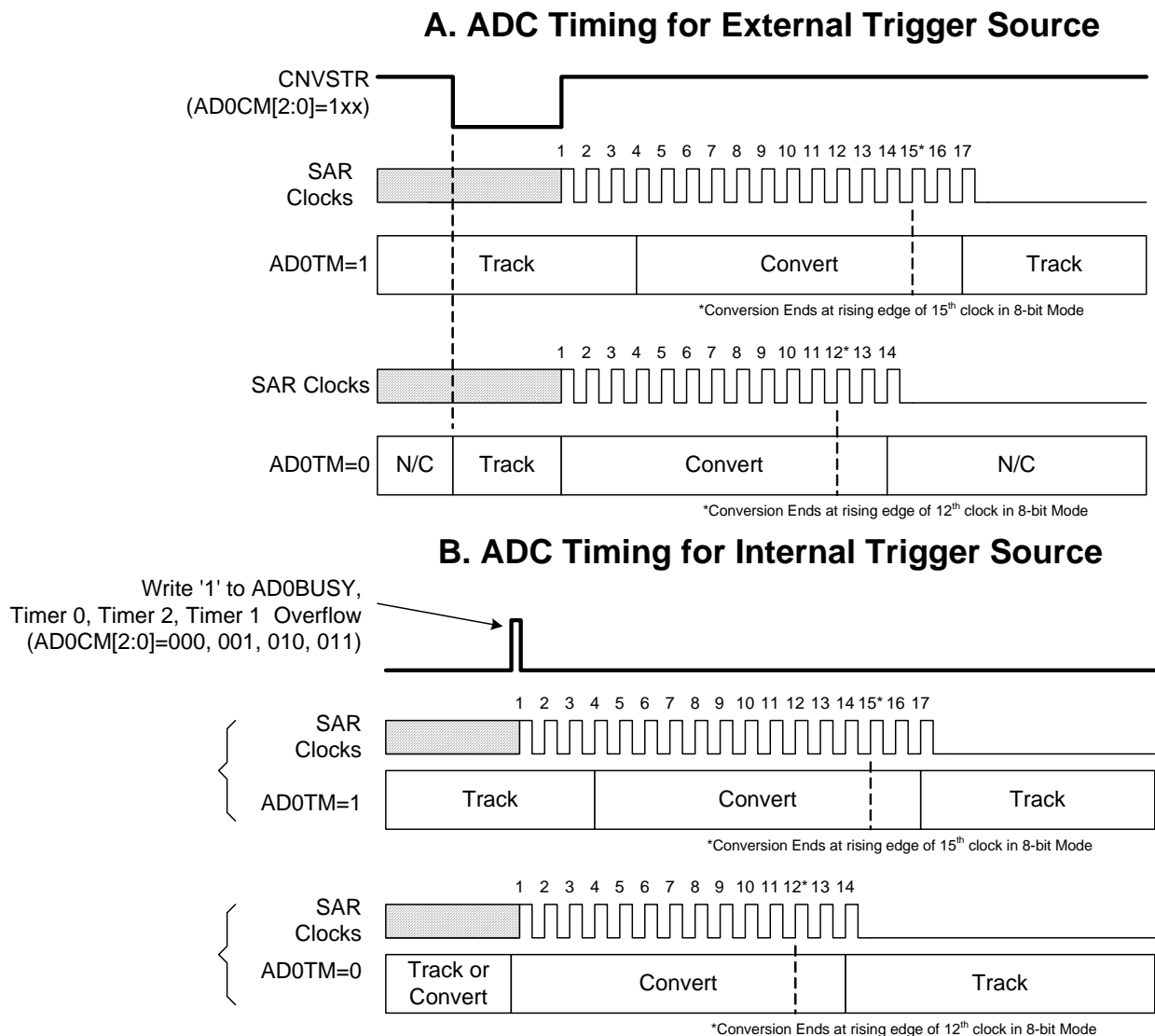
1. Writing a 1 to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 1 overflow
5. A rising edge on the CNVSTR input signal
6. A Timer 3 overflow

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. When Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See Section "33. Timers" on page 262 for timer configuration.

**Important Note About Using CNVSTR:** The CNVSTR input pin also functions as a Port I/O pin. When the CNVSTR input is used as the ADC0 conversion source, the associated pin should be skipped by the Digital Crossbar. See Section "28. Port Input/Output" on page 180 for details on Port I/O configuration.

## 10.3.2. Tracking Modes

The AD0TM bit in register ADC0CN enables "delayed conversions", and will delay the actual conversion start by three SAR clock cycles, during which time the ADC will continue to track the input. If AD0TM is left at logic 0, a conversion will begin immediately, without the extra tracking time. For internal start-of-conversion sources, the ADC will track anytime it is not performing a conversion. When the CNVSTR signal is used to initiate conversions, ADC0 will track either when AD0TM is logic 1, or when AD0TM is logic 0 and CNVSTR is held low. See Figure 10.2 for track and convert timing details. Delayed conversion mode is useful when AMUX settings are frequently changed, due to the settling time requirements described in Section "10.3.3. Settling Time Requirements" on page 58.



**Figure 10.2. 10-Bit ADC Track and Conversion Example Timing**

# C8051F70x/71x

## SFR Definition 14.2. CPT0MD: Comparator0 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP0RIE	CP0FIE			CP0MD[1:0]	
Type	R	R	R/W	R/W	R	R	R/W	
Reset	0	0	0	0	0	0	1	0

SFR Address = 0x9D; SFR Page = 0

Bit	Name	Function
7:6	Unused	Read = 00b, Write = Don't Care.
5	CP0RIE	<b>Comparator0 Rising-Edge Interrupt Enable.</b> 0: Comparator0 Rising-edge interrupt disabled. 1: Comparator0 Rising-edge interrupt enabled.
4	CP0FIE	<b>Comparator0 Falling-Edge Interrupt Enable.</b> 0: Comparator0 Falling-edge interrupt disabled. 1: Comparator0 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP0MD[1:0]	<b>Comparator0 Mode Select.</b> These bits affect the response time and power consumption for Comparator0. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

## 15. Capacitive Sense (CS0)

The Capacitive Sense subsystem uses a capacitance-to-digital circuit to determine the capacitance on a port pin. The module can take measurements from different port pins using the module's analog multiplexer. The module is enabled only when the CS0EN bit (CS0CN) is set to 1. Otherwise the module is in a low-power shutdown state. The module can be configured to take measurements on one port pin or a group of port pins, using auto-scan. A selectable gain circuit allows the designer to adjust the maximum allowable capacitance. An accumulator is also included, which can be configured to average multiple conversions on an input channel. Interrupts can be generated when CS0 completes a conversion or when the measured value crosses a threshold defined in CS0THH:L.

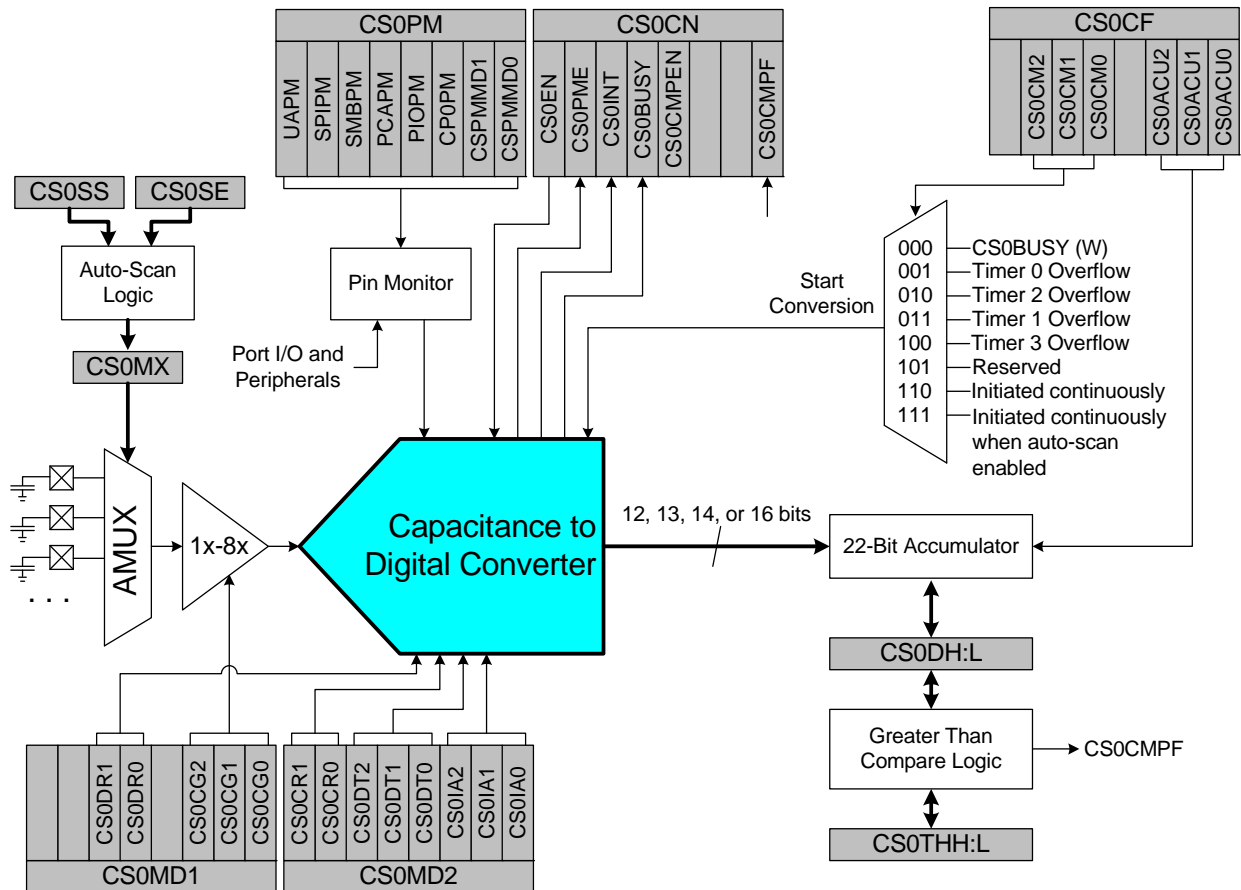


Figure 15.1. CS0 Block Diagram

**SFR Definition 15.10. CS0MD1: Capacitive Sense Mode 1**

Bit	7	6	5	4	3	2	1	0
Name			CS0DR[1:0]			CS0CG[2:0]		
Type	R		R/W		R	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	1

SFR Address = 0xAD; SFR Page = 0

Bit	Name	Description
7:6	Unused	Read = 00b; Write = Don't care
5:4	CS0DR[1:0]	<b>CS0 Double Reset Select.</b> These bits adjust the secondary CS0 reset time. For most touch-sensitive switches, the default (fastest) value is sufficient, and these bits should not be modified.  00: No additional time is used for secondary reset (recommended for most switches)  01: An additional 0.75 $\mu$ s is used for secondary reset.  10: An additional 1.5 $\mu$ s is used for secondary reset.  11: An additional 2.25 $\mu$ s is used for secondary reset.
3	Unused	Read = 0b; Write = Don't care
2:0	CS0CG[2:0]	<b>CS0 Reference Gain Select.</b> These bits select the "gain" applied to the current used to charge an internal reference capacitor. Lower gain values decrease the current setting, and increase both the size of the capacitance that can be measured with the CS0 module, and the base conversion time. Refer to "15.2. CS0 Gain Adjustment" on page 81 for more information.  000: Gain = 1x 001: Gain = 2x 010: Gain = 3x 011: Gain = 4x 100: Gain = 5x 101: Gain = 6x 110: Gain = 7x 111: Gain = 8x (default)

## SFR Definition 18.1. EMI0CN: External Memory Interface Control

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

SFR Address = 0xAA; SFR Page = F

Bit	Name	Function
7:0	PGSEL[7:0]	<b>XRAM Page Select Bits.</b> The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. 0x00: 0x0000 to 0x00FF 0x01: 0x0100 to 0x01FF ... 0xFE: 0xFE00 to 0xFEFF 0xFF: 0xFF00 to 0xFFFF



---

## 21.1. MCU Interrupt Sources and Vectors

The C8051F70x/71x MCUs support 16 interrupt sources. Software can simulate an interrupt by setting an interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 21.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

### 21.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 21.1.

### 21.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

---

**SFR Definition 21.2. IP: Interrupt Priority**


---

Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0xB8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1b, Write = Don't Care.
6	PSPI0	<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.

## 21.3. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ External Interrupts

The  $\overline{\text{INT0}}$  and  $\overline{\text{INT1}}$  external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ( $\overline{\text{INT0}}$  Polarity) and IN1PL ( $\overline{\text{INT1}}$  Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “33.1. Timer 0 and Timer 1” on page 264) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt	IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive	1	0	Active low, edge sensitive
1	1	Active high, edge sensitive	1	1	Active high, edge sensitive
0	0	Active low, level sensitive	0	0	Active low, level sensitive
0	1	Active high, level sensitive	0	1	Active high, level sensitive

$\overline{\text{INT0}}$  and  $\overline{\text{INT1}}$  are assigned to Port pins as defined in the IT01CF register (see SFR Definition 21.7). Note that  $\overline{\text{INT0}}$  and  $\overline{\text{INT1}}$  Port pin assignments are independent of any Crossbar assignments.  $\overline{\text{INT0}}$  and  $\overline{\text{INT1}}$  will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to  $\overline{\text{INT0}}$  and/or  $\overline{\text{INT1}}$ , configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “28.3. Priority Crossbar Decoder” on page 185 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the  $\overline{\text{INT0}}$  and  $\overline{\text{INT1}}$  external interrupts, respectively. If an  $\overline{\text{INT0}}$  or  $\overline{\text{INT1}}$  external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

**SFR Definition 21.7. IT01CF: INT0/INT1 Configuration**

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

SFR Address = 0xE4; SFR Page = F

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

---

## 27.3. 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 27.1. A 10 M $\Omega$  resistor also must be wired across the XTAL2 and XTAL1 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 27.1. 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 27.4).

**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 P0.2 and P0.3 are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “28.3. Priority Crossbar Decoder” on page 185 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 “28.4. Port I/O Initialization” on page 189 for details on Port input mode selection.

**SFR Definition 30.1. SMB0CF: SMBus Clock/Configuration**

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0

Bit	Name	Function
7	ENSMB	<b>SMBus Enable.</b> This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	<b>SMBus Slave Inhibit.</b> When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	<b>SMBus Busy Indicator.</b> This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	<b>SMBus Setup and Hold Time Extension Enable.</b> This bit controls the SDA setup and hold times according to Table 30.2. 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	<b>SMBus SCL Timeout Detection Enable.</b> This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	<b>SMBus Free Timeout Detection Enable.</b> When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS[1:0]	<b>SMBus Clock Source Selection.</b> These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 30.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

# C8051F70x/71x

## SFR Definition 30.4. SMB0ADM: SMBus Slave Address Mask

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

SFR Address = 0xBB; SFR Page = F

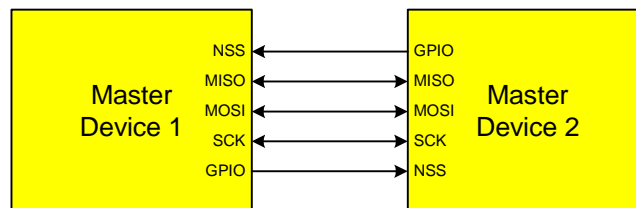
Bit	Name	Function
7:1	SLVM[6:0]	<b>SMBus Slave Address Mask.</b> Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM[6:0] enables comparisons with the corresponding bit in SLV[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK	<b>Hardware Acknowledge Enable.</b> Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled.

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.

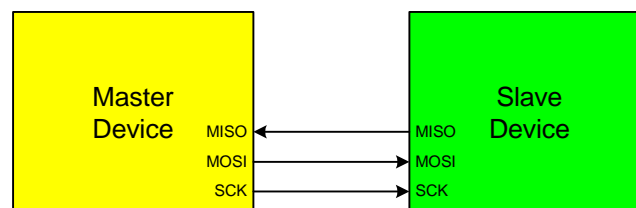
When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is 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 31.2 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 31.3 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 31.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

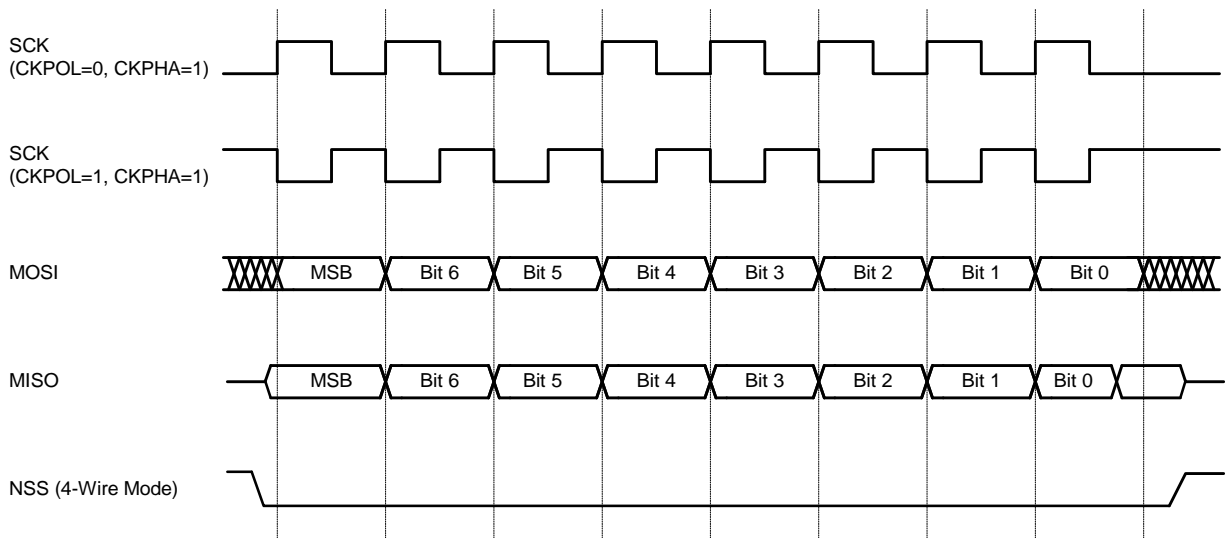


**Figure 31.2. Multiple-Master Mode Connection Diagram**



**Figure 31.3. 3-Wire Single Master and Single Slave Mode Connection Diagram**





**Figure 31.7. Slave Mode Data/Clock Timing (CKPHA = 1)**

### 31.6. 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.

## 32.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

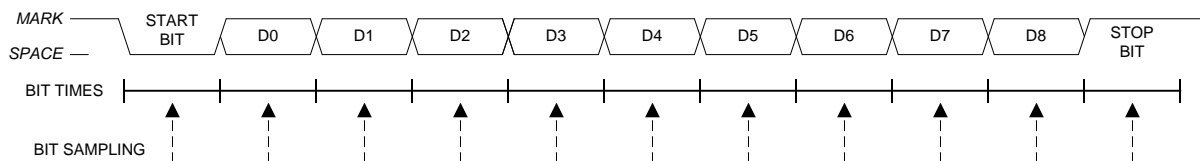
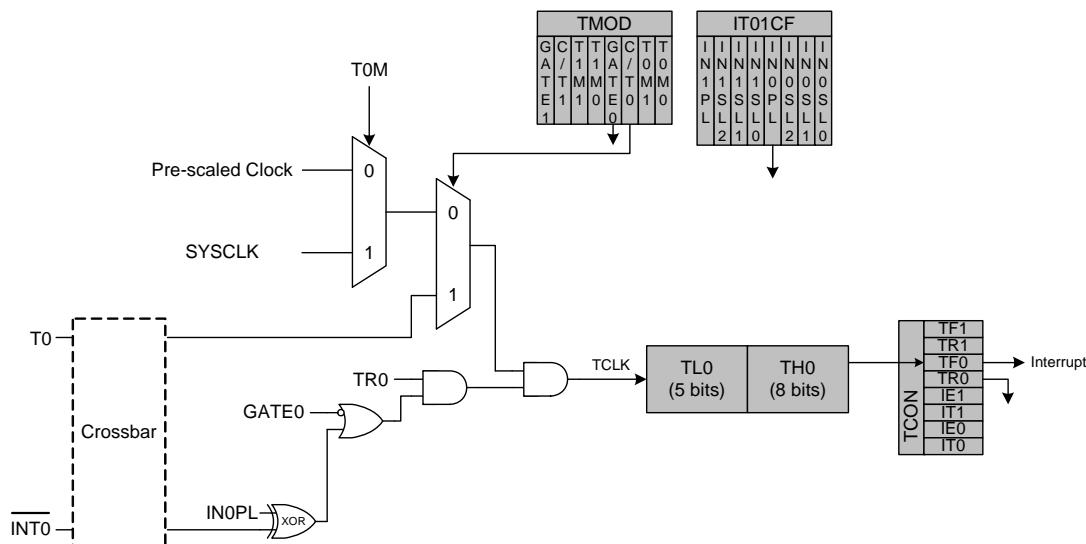


Figure 32.5. 9-Bit UART Timing Diagram



**Figure 33.1. T0 Mode 0 Block Diagram**

## 33.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

## 33.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see Section “21.3. INT0 and INT1 External Interrupts” on page 146 for details on the external input signals INT0 and INT1).

# C8051F70x/71x

## SFR Definition 33.16. TMR3L: Timer 3 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x94; SFR Page = 0

Bit	Name	Function
7:0	TMR3L[7:0]	<b>Timer 3 Low Byte.</b> In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value.

## SFR Definition 33.17. TMR3H Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x95; SFR Page = 0

Bit	Name	Function
7:0	TMR3H[7:0]	<b>Timer 3 High Byte.</b> 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.

**SFR Definition 34.4. PCA0CPMn: PCA Capture/Compare Mode**

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPM0 = 0xDA, PCA0CPM1 = 0xDB, PCA0CPM2 = 0xDC

SFR Pages: PCA0CPM0 = F, PCA0CPM1 = F, PCA0CPM2 = F

Bit	Name	Function
7	PWM16n	<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	ECOMn	<b>Comparator Function Enable.</b> This bit enables the comparator function for PCA module n when set to 1.
5	CAPPn	<b>Capture Positive Function Enable.</b> This bit enables the positive edge capture for PCA module n when set to 1.
4	CAPNn	<b>Capture Negative Function Enable.</b> This bit enables the negative edge capture for PCA module n when set to 1.
3	MATn	<b>Match Function Enable.</b> This bit enables the match function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	<b>Toggle Function Enable.</b> This bit enables the toggle function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	<b>Pulse Width Modulation Mode Enable.</b> This bit enables the PWM function for PCA module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	ECCFn	<b>Capture/Compare Flag Interrupt Enable.</b> This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.