

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	SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	29
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f310">https://www.e-xfl.com/product-detail/silicon-labs/c8051f310</a>

# C8051F310/1/2/3/4/5/6/7

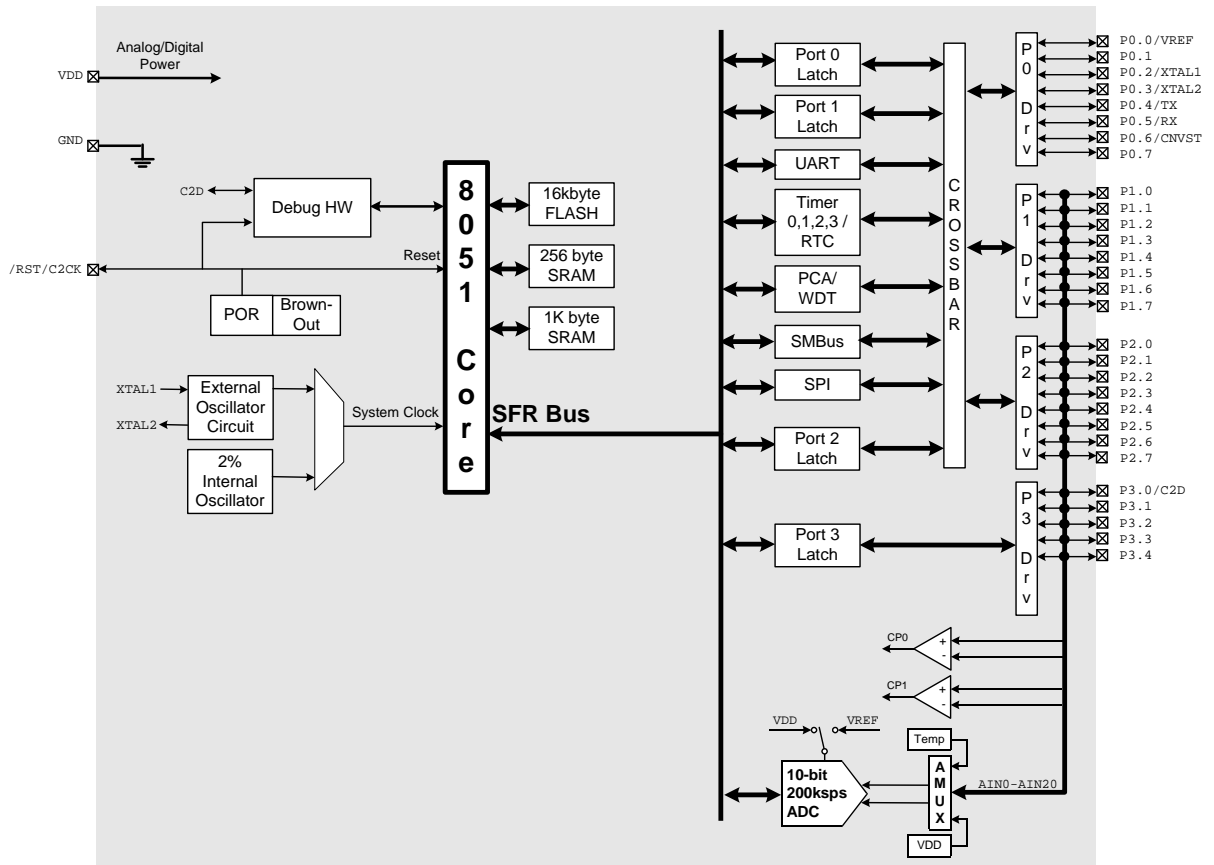


Figure 1.1. C8051F310 Block Diagram

## 1.1. CIP-51™ Microcontroller Core

### 1.1.1. Fully 8051 Compatible

The C8051F31x family utilizes Silicon Laboratories' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The CIP-51 core offers all the peripherals included with a standard 8052, including four 16-bit counter/timers, a full-duplex UART with extended baud rate configuration, an enhanced SPI port, 1280 bytes of internal RAM, 128 byte Special Function Register (SFR) address space, and 29/25/21 I/O pins.

### 1.1.2. Improved Throughput

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute with a maximum system clock of 12-to-24 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with only four instructions taking more than four system clock cycles.

The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. Figure 1.9 shows a comparison of peak throughputs for various 8-bit microcontroller cores with their maximum system clocks.

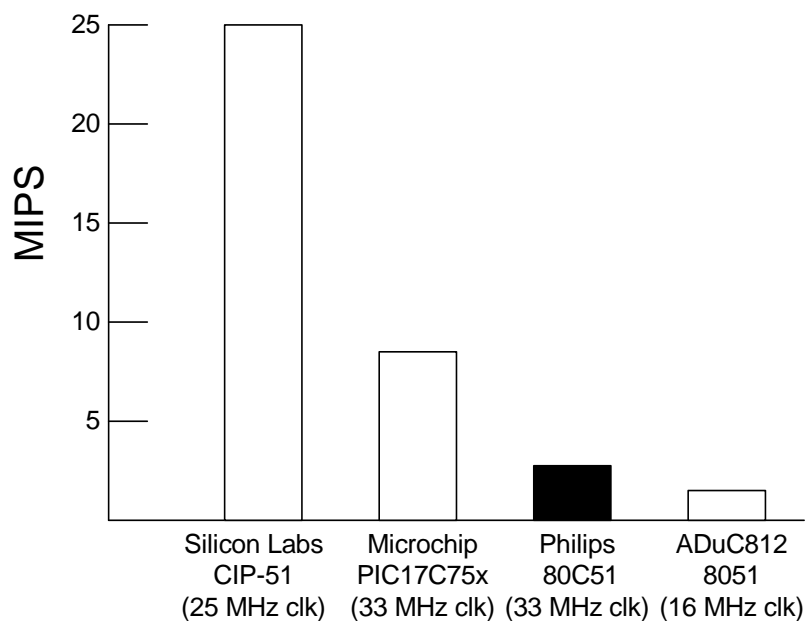


Figure 1.9. Comparison of Peak MCU Execution Speeds

## 5.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

### SFR Definition 5.7. ADC0GTH: ADC0 Greater-Than Data High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC4

Bits7–0: High byte of ADC0 Greater-Than Data Word.

### SFR Definition 5.8. ADC0GTL: ADC0 Greater-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC3

Bits7–0: Low byte of ADC0 Greater-Than Data Word.

## SFR Definition 7.2. CPT0MX: Comparator0 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CMX0N1	CMX0N0	-	-	CMX0P1	CMX0P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

Bits7–6: UNUSED. Read = 00b, Write = don't care.

Bits5–4: CMX0N1–CMX0N0: Comparator0 Negative Input MUX Select.

These bits select which Port pin is used as the Comparator0 negative input.

CMX0N1	CMX0N0	Negative Input
0	0	P1.1
0	1	P1.5
1	0	P2.1
1	1	P2.5

Bits3–2: UNUSED. Read = 00b, Write = don't care.

Bits1–0: CMX0P1–CMX0P0: Comparator0 Positive Input MUX Select.

These bits select which Port pin is used as the Comparator0 positive input.

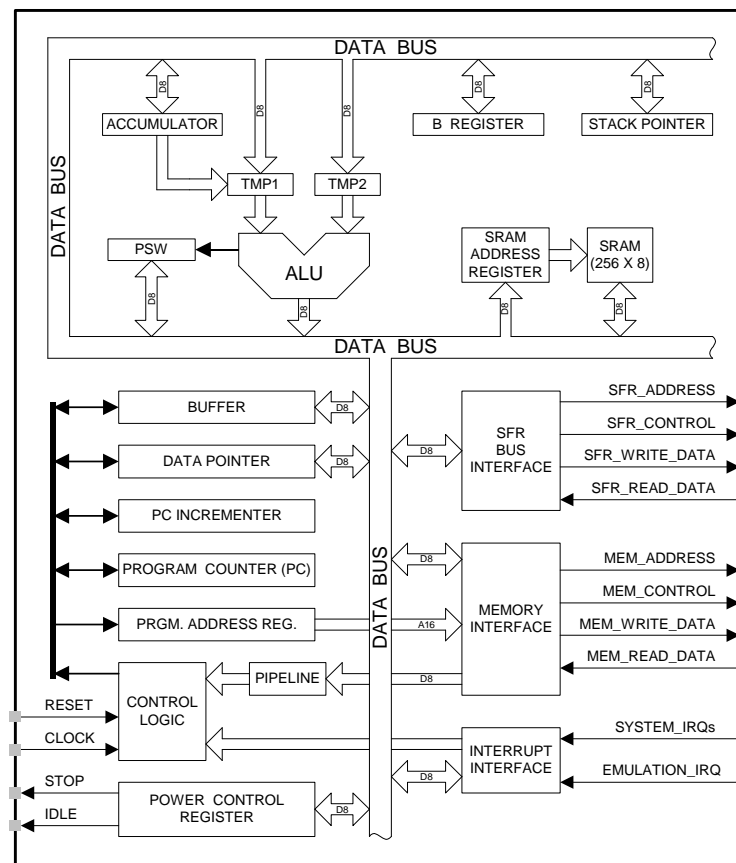
CMX0P1	CMX0P0	Positive Input
0	0	P1.0
0	1	P1.4
1	0	P2.0
1	1	P2.4

## 8. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. Included are four 16-bit counter/timers (see description in **Section 17**), an enhanced full-duplex UART (see description in **Section 15**), an Enhanced SPI (see description in **Section 16**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (**Section 8.2.6**), and 29 Port I/O (see description in **Section 13**). The CIP-51 also includes on-chip debug hardware (see description in **Section 20**), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 8.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- 256 Bytes of Internal RAM
- 29 Port I/O
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security



**Figure 8.1. CIP-51 Block Diagram**

# C8051F310/1/2/3/4/5/6/7

10. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

## 10.4.3. System Clock

12. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in AN201, "Writing to Flash from Firmware", available from the Silicon Laboratories web site.

### SFR Definition 10.1. PSCTL: Program Store R/W Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8F

Bits7–2: UNUSED: Read = 000000b, Write = don't care.

Bit1: PSEE: Program Store Erase Enable  
Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter.  
0: Flash program memory erasure disabled.  
1: Flash program memory erasure enabled.

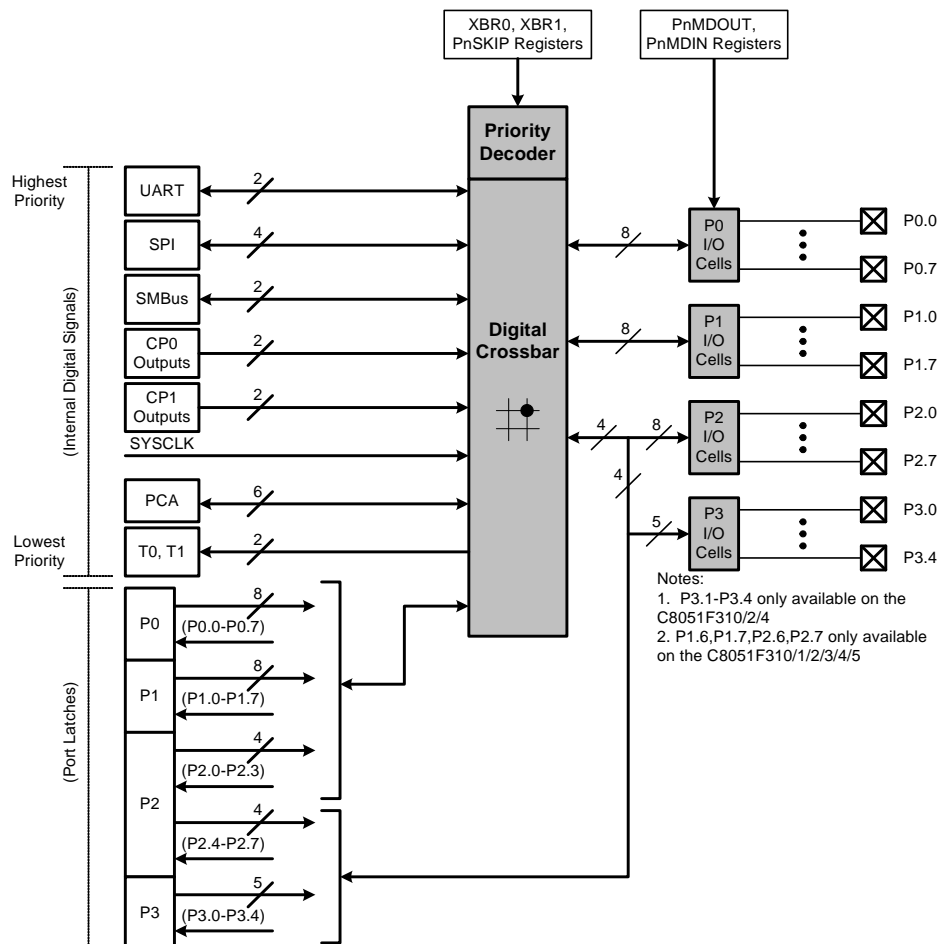
Bit0: PSWE: Program Store Write Enable  
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The Flash location should be erased before writing data.  
0: Writes to Flash program memory disabled.  
1: Writes to Flash program memory enabled; the MOVX write instruction targets Flash memory.

## 13. Port Input/Output

Digital and analog resources are available through 29 I/O pins (C8051F310/2/4), or 25 I/O pins (C8051F311/3/5), or 21 I/O pins (C8051F316/7). Port pins are organized as three byte-wide Ports and one 5-bit-wide (C8051F310/2/4) or 1-bit-wide (C8051F311/3/5) Port. In the C8051F316/7, the port pins are organized as one byte-wide Port, two 6-bit-wide Ports and one 1-bit-wide Port. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input; Port pins P0.0-P2.3 can be assigned to one of the internal digital resources as shown in Figure 13.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. The state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the Priority Decoder (Figure 13.3 and Figure 13.4). The registers XBR0 and XBR1, defined in SFR Definition 13.1 and SFR Definition 13.2, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 13.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1,2,3). Complete Electrical Specifications for Port I/O are given in Table 13.1 on page 143.



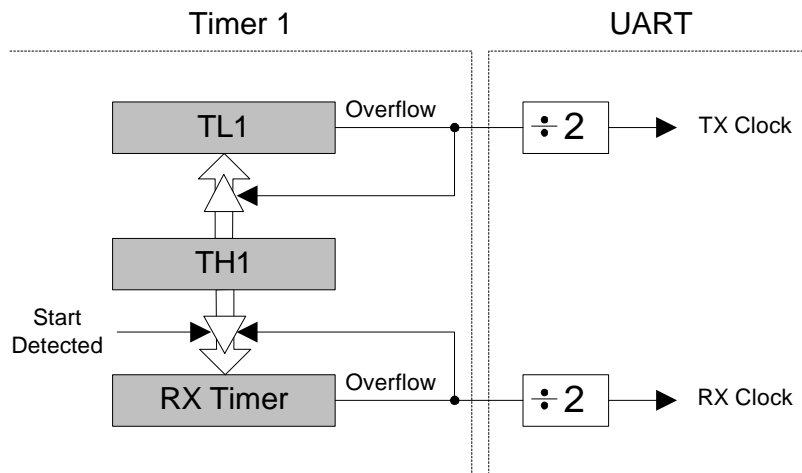
**Figure 13.1. Port I/O Functional Block Diagram**

## SFR Definition 13.1. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE1
<p>Bit7: CP1AE: Comparator1 Asynchronous Output Enable 0: Asynchronous CP1 unavailable at Port pin. 1: Asynchronous CP1 routed to Port pin.</p> <p>Bit6: CP1E: Comparator1 Output Enable 0: CP1 unavailable at Port pin. 1: CP1 routed to Port pin.</p> <p>Bit5: CP0AE: Comparator0 Asynchronous Output Enable 0: Asynchronous CP0 unavailable at Port pin. 1: Asynchronous CP0 routed to Port pin.</p> <p>Bit4: CP0E: Comparator0 Output Enable 0: CP0 unavailable at Port pin. 1: CP0 routed to Port pin.</p> <p>Bit3: SYSCKE: /SYSCLK Output Enable 0: /SYSCLK unavailable at Port pin. 1: /SYSCLK output routed to Port pin.</p> <p>Bit2: SMB0E: SMBus I/O Enable 0: SMBus I/O unavailable at Port pins. 1: SMBus I/O routed to Port pins.</p> <p>Bit1: SPI0E: SPI I/O Enable 0: SPI I/O unavailable at Port pins. 1: SPI I/O routed to Port pins.</p> <p>Bit0: URT0E: UART I/O Output Enable 0: UART I/O unavailable at Port pin. 1: UART TX0, RX0 routed to Port pins P0.4 and P0.5.</p>								

## 15.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 15.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.



**Figure 15.2. UART0 Baud Rate Logic**

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see **Section “17.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 189**). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK / 4, SYSCLK / 12, SYSCLK / 48, the external oscillator clock / 8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 15.1.

### Equation 15.1. UART0 Baud Rate

$$UartBaudRate = \frac{T1_{CLK}}{(256 - T1H)} \times \frac{1}{2}$$

Where  $T1_{CLK}$  is the frequency of the clock supplied to Timer 1, and  $T1H$  is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in **Section “17. Timers” on page 187**. A quick reference for typical baud rates and system clock frequencies is given in Table 15.1 through Table 15.6. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

## SFR Definition 15.1. SCON0: Serial Port 0 Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S0MODE		MCE0	REN0	TB80	RB80	TI0	RI0	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable
SFR Address: 0x98								
<p>Bit7: S0MODE: Serial Port 0 Operation Mode. This bit selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate. 1: 9-bit UART with Variable Baud Rate.</p> <p>Bit6: UNUSED. Read = 1b. Write = don't care.</p> <p>Bit5: MCE0: Multiprocessor Communication Enable. The function of this bit is dependent on the Serial Port 0 Operation Mode. S0MODE = 0: Checks for valid stop bit. 0: Logic level of stop bit is ignored. 1: RI0 will only be activated if stop bit is logic level 1. S0MODE = 1: Multiprocessor Communications Enable. 0: Logic level of ninth bit is ignored. 1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.</p> <p>Bit4: REN0: Receive Enable. This bit enables/disables the UART receiver. 0: UART0 reception disabled. 1: UART0 reception enabled.</p> <p>Bit3: TB80: Ninth Transmission Bit. The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.</p> <p>Bit2: RB80: Ninth Receive Bit. RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.</p> <p>Bit1: TI0: Transmit Interrupt Flag. Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.</p> <p>Bit0: RI0: Receive Interrupt Flag. Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.</p>								

---

## 16.1. Signal Descriptions

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

### 16.1.1. 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.

### 16.1.2. 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.

### 16.1.3. 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.

### 16.1.4. 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:

- 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.
- 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.
- 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.

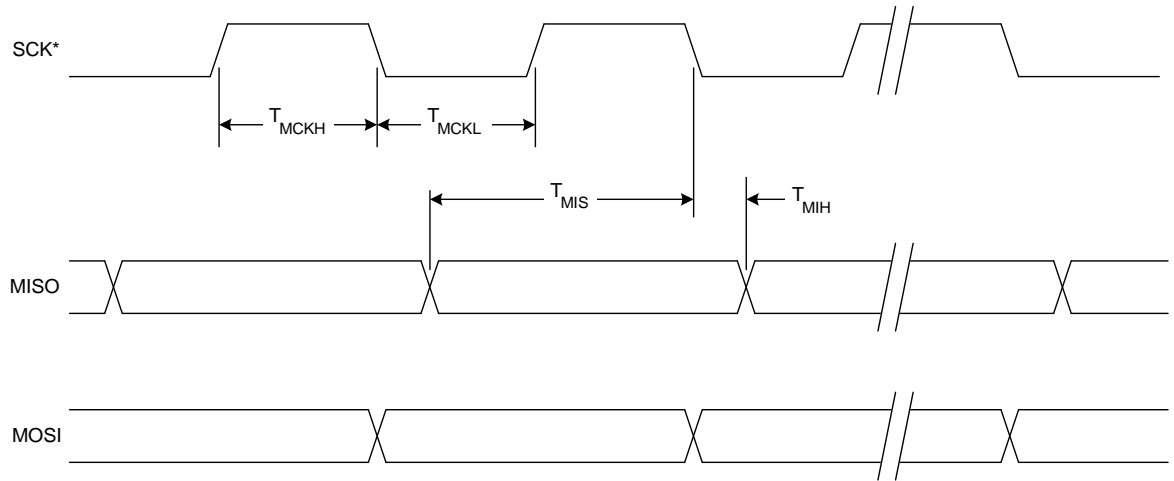
See Figure 16.2, Figure 16.3, and Figure 16.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “13. Port Input/Output” on page 129 for general purpose port I/O and crossbar information.

## 16.6. 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 register definitions.

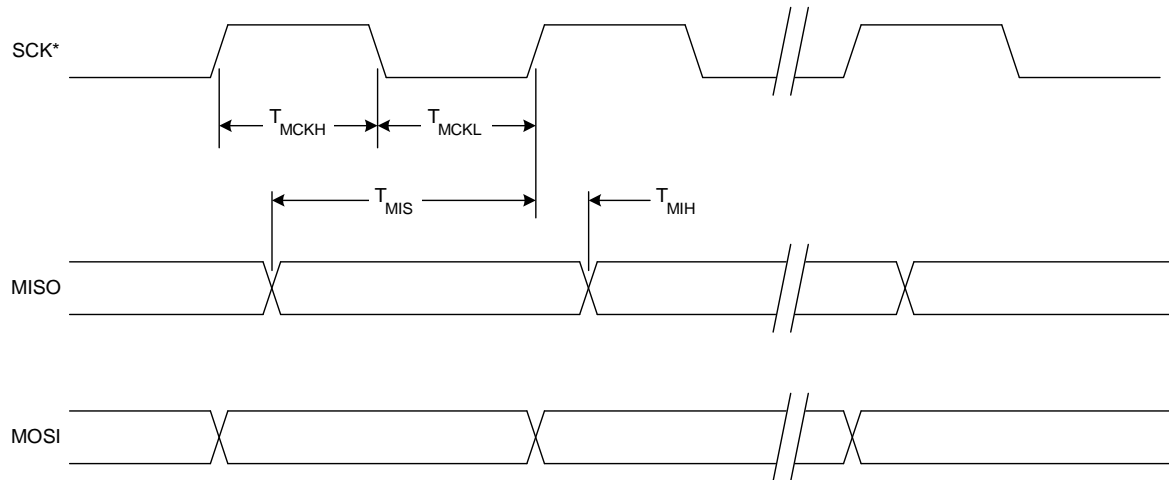
### SFR Definition 16.1. SPI0CFG: SPI0 Configuration

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SFR Address: 0xA1								
Bit 7:	SPIBSY: SPI Busy (read only). This bit is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).							
Bit 6:	MSTEN: Master Mode Enable. 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.							
Bit 5:	CKPHA: SPI0 Clock Phase. This bit controls the SPI0 clock phase. 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*							
Bit 4:	CKPOL: SPI0 Clock Polarity. This bit controls the SPI0 clock polarity. 0: SCK line low in idle state. 1: SCK line high in idle state.							
Bit 3:	SLVSEL: Slave Selected Flag (read only). 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.							
Bit 2:	NSSIN: NSS Instantaneous Pin Input (read only). 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.							
Bit 1:	SRMT: Shift Register Empty (Valid in Slave Mode, read only). 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. NOTE: SRMT = 1 when in Master Mode.							
Bit 0:	RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only). This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. NOTE: RXBMT = 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 SYSCLOCK before the end of each data bit, to provide maximum settling time for the slave device. See Table 16.1 for timing parameters.							



\* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 16.8. SPI Master Timing (CKPHA = 0)**



\* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 16.9. SPI Master Timing (CKPHA = 1)**

## 17. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M-T0M) and the Clock Scale bits (SCA1-SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 17.3 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

### 17.1. Timer 0 and Timer 1

Each timer is implemented as 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (SFR Definition 8.7. "IE: Interrupt Enable" on page 97); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1-T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

#### 17.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

# C8051F310/1/2/3/4/5/6/7

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to **Section “13.1. Priority Crossbar Decoder” on page 131** for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 17.3).

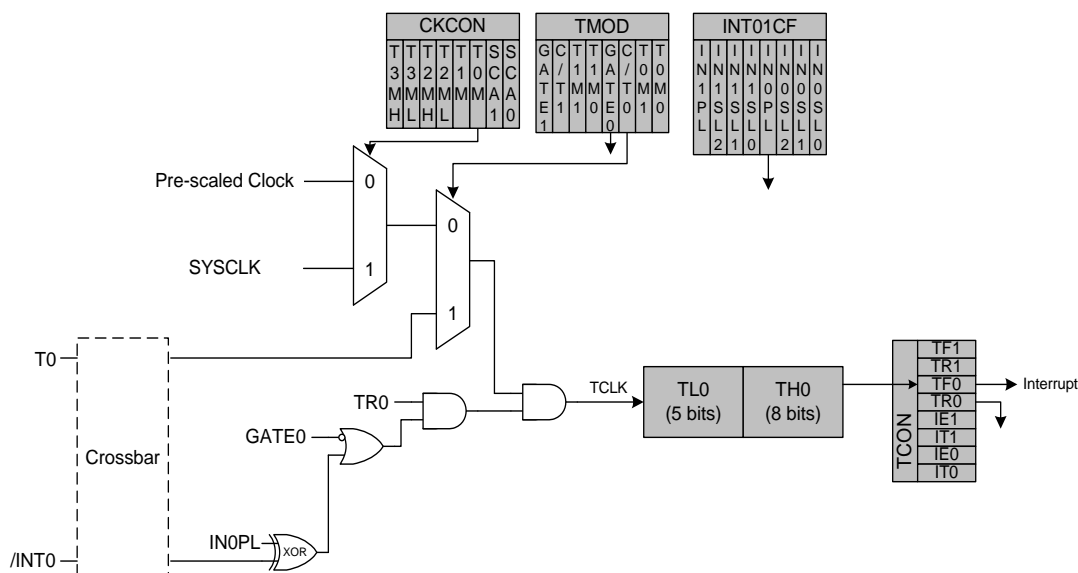
Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 8.11. “IT01CF: INT0/INT1 Configuration” on page 101). Setting GATE0 to ‘1’ allows the timer to be controlled by the external input signal /INT0 (see **Section “8.3.5. Interrupt Register Descriptions” on page 97**), facilitating pulse width measurements.

TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

X = Don't Care

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal /INT1 is used with Timer 1; the /INT1 polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 8.11. “IT01CF: INT0/INT1 Configuration” on page 101).



**Figure 17.1. T0 Mode 0 Block Diagram**

# C8051F310/1/2/3/4/5/6/7

## 17.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 17.5. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bit (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

Note: External clock divided by 8 is synchronized with the system clock, and the external clock must be less than or equal to the system clock to operate in this mode.

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

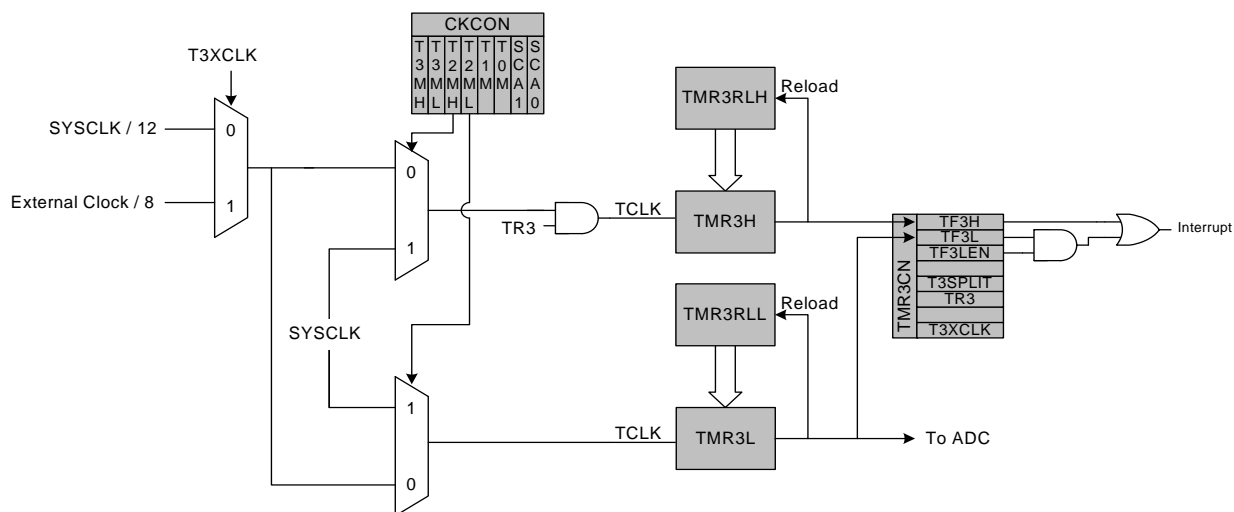


Figure 17.7. Timer 3 8-Bit Mode Block Diagram

# C8051F310/1/2/3/4/5/6/7

## SFR Definition 17.14. TMR3RLL: Timer 3 Reload Register Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x92

Bits 7–0: TMR3RLL: Timer 3 Reload Register Low Byte.  
TMR3RLL holds the low byte of the reload value for Timer 3.

## SFR Definition 17.15. TMR3RLH: Timer 3 Reload Register High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x93

Bits 7–0: TMR3RLH: Timer 3 Reload Register High Byte.  
The TMR3RLH holds the high byte of the reload value for Timer 3.

## SFR Definition 17.16. TMR3L: Timer 3 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x94

Bits 7–0: TMR3L: Timer 3 Low Byte.  
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 17.17. TMR3H Timer 3 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x95

Bits 7–0: TMR3H: Timer 3 High Byte.  
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.



Note that the 8-bit offset held in PCA0CPH4 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 18.4, where PCA0L is the value of the PCA0L register at the time of the update.

## Equation 18.4. Watchdog Timer Offset in PCA Clocks

$$Offset = (256 \times PCA0CPL4) + (256 - PCA0L)$$

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH4 and PCA0H. Software may force a WDT reset by writing a '1' to the CCF4 flag (PCA0CN.4) while the WDT is enabled.

### 18.3.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a '0' to the WDTE bit.
- Select the desired PCA clock source (with the CPS2-CPS0 bits).
- Load PCA0CPL4 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to '1'.
- Write a value to PCA0CPH4 to reload the WDT.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL4 defaults to 0x00. Using Equation 18.4, this results in a WDT timeout interval of 256 system clock cycles. Table 18.3 lists some example timeout intervals for typical system clocks.

---

## 19.3. PCA Counter

On “REV A” devices, if the main PCA counter (PCA0H : PCA0L) overflows during the execution phase of a read-modify-write instruction (bit-wise SETB or CLR, ANL, ORL, XRL) that targets the PCA0CN register, the CF (Counter Overflow) bit will not be set. An example software work-around is as follows:

- Step 1. Disable global interrupts (EA = 0).
- Step 2. Read PCA0L. This will latch the value of PCA0H.
- Step 3. Read PCA0H, saving the value.
- Step 4. Execute the bit-wise operation on CCFn (for example, CLR CCF0, or CCF0 = 0;).
- Step 5. Read PCA0L.
- Step 6. Read PCA0H, saving the value.
- Step 7. If the value of PCA0H read in Step 3 is 0xFF and the value for PCA0H read in Step 6 is 0x00, then manually set the CF bit in software (for example, SETB CF, or CF = 1;).
- Step 8. Re-enable interrupts (EA = 1).

This behavior is not present on “REV B” and later devices. Software written for “REV A” devices will run on “REV B” and later devices without modification.