

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

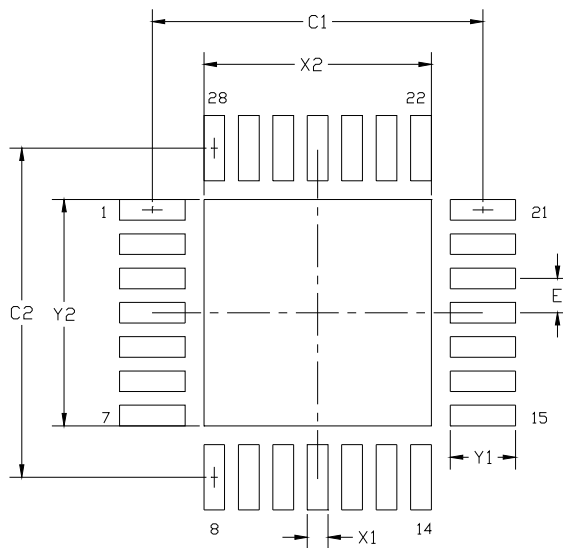
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	21
Program Memory Size	16KB (16K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051t616-gm">https://www.e-xfl.com/product-detail/silicon-labs/c8051t616-gm</a>

---

22.3.2. Arbitration.....	134
22.3.3. Clock Low Extension.....	134
22.3.4. SCL Low Timeout.....	134
22.3.5. SCL High (SMBus Free) Timeout .....	135
22.4. Using the SMBus.....	135
22.4.1. SMBus Configuration Register.....	135
22.4.2. SMB0CN Control Register .....	139
22.4.3. Data Register .....	142
22.5. SMBus Transfer Modes.....	143
22.5.1. Write Sequence (Master) .....	143
22.5.2. Read Sequence (Master).....	144
22.5.3. Write Sequence (Slave) .....	145
22.5.4. Read Sequence (Slave).....	146
22.6. SMBus Status Decoding.....	146
<b>23. UART0.....</b>	<b>149</b>
23.1. Enhanced Baud Rate Generation.....	150
23.2. Operational Modes .....	151
23.2.1. 8-Bit UART.....	151
23.2.2. 9-Bit UART .....	152
23.3. Multiprocessor Communications .....	153
<b>24. Enhanced Serial Peripheral Interface (SPI0) .....</b>	<b>157</b>
24.1. Signal Descriptions.....	158
24.1.1. Master Out, Slave In (MOSI).....	158
24.1.2. Master In, Slave Out (MISO).....	158
24.1.3. Serial Clock (SCK) .....	158
24.1.4. Slave Select (NSS) .....	158
24.2. SPI0 Master Mode Operation.....	159
24.3. SPI0 Slave Mode Operation.....	160
24.4. SPI0 Interrupt Sources .....	161
24.5. Serial Clock Phase and Polarity .....	161
24.6. SPI Special Function Registers.....	163
<b>25. Timers .....</b>	<b>170</b>
25.1. Timer 0 and Timer 1 .....	172
25.1.1. Mode 0: 13-bit Counter/Timer .....	172
25.1.2. Mode 1: 16-bit Counter/Timer .....	173
25.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload.....	174
25.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only).....	175
25.2. Timer 2 .....	180
25.2.1. 16-bit Timer with Auto-Reload.....	180
25.2.2. 8-bit Timers with Auto-Reload.....	181
25.3. Timer 3 .....	185
25.3.1. 16-bit Timer with Auto-Reload.....	185
25.3.2. 8-bit Timers with Auto-Reload.....	186
<b>26. Programmable Counter Array.....</b>	<b>190</b>
26.1. PCA Counter/Timer .....	191

---

# C8051T610/1/2/3/4/5/6/7



**Figure 5.2. QFN-28 Recommended PCB Land Pattern**

**Table 5.2. QFN-28 PCB Land Pattern Dimensions**

Dimension	Min	Max	Dimension	Min	Max
C1	4.80		X2	3.20	3.30
C2	4.80		Y1	0.85	0.95
E	0.50		Y2	3.20	3.30
X1	0.20	0.30			

**Notes:**

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

4. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60µm minimum, all the way around the pad.

Stencil Design

5. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
6. The stencil thickness should be 0.125mm (5 mils).
7. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pins.
8. A 3x3 array of 0.90mm openings on a 1.1mm pitch should be used for the center pad to assure the proper paste volume.

Card Assembly

9. A No-Clean, Type-3 solder paste is recommended.
10. The recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

# C8051T610/1/2/3/4/5/6/7

## 7.2. Electrical Characteristics

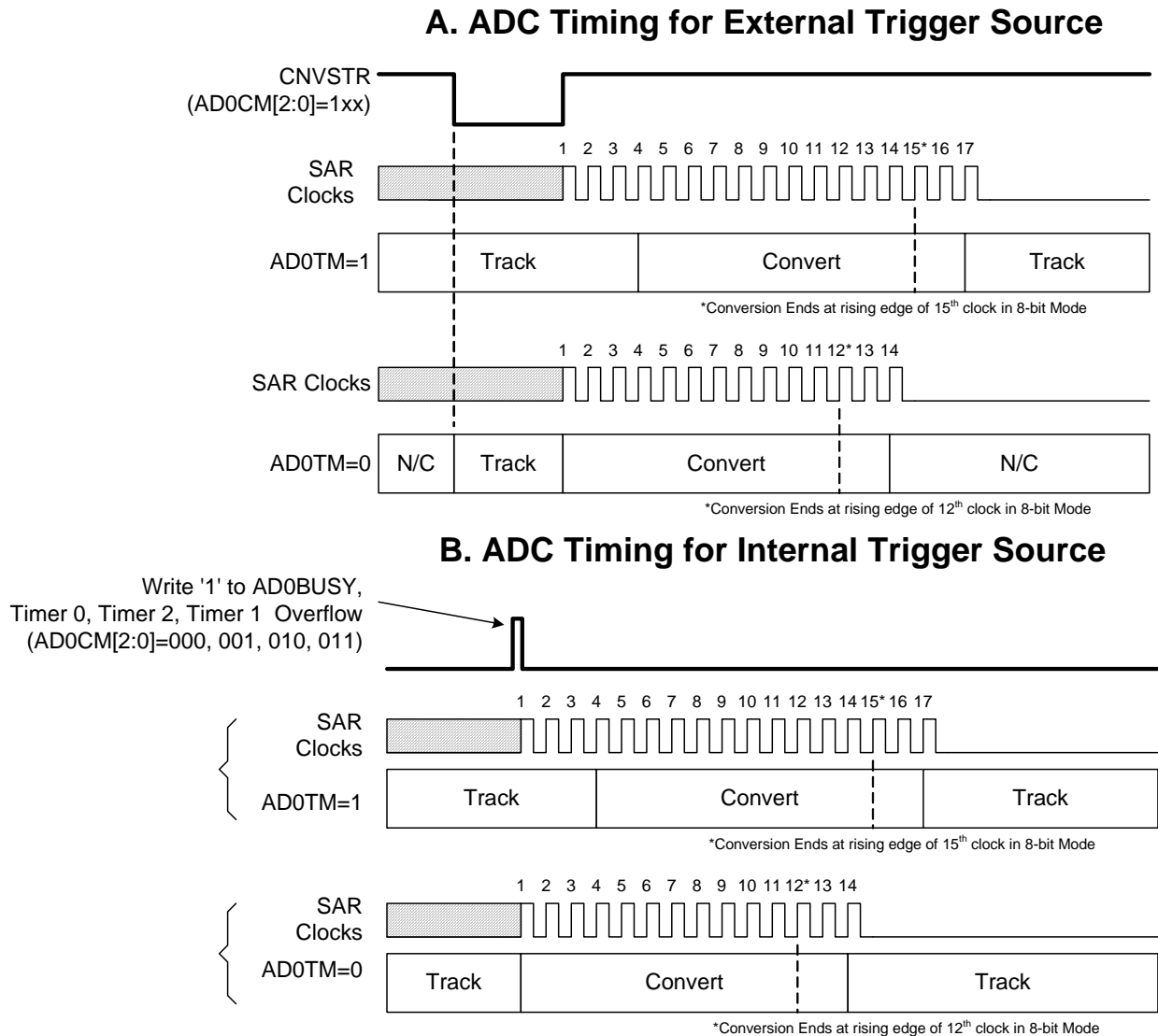
**Table 7.2. Global Electrical Characteristics**

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Supply Voltage (Note 1)	Regulator in Normal Mode	1.8	3.0	3.6	V
	Regulator in Bypass Mode	1.7	1.8	1.9	V
Digital Supply Current with CPU Active	V <sub>DD</sub> = 1.8 V, Clock = 25 MHz	—	6.2	8.8	mA
	V <sub>DD</sub> = 1.8 V, Clock = 1 MHz	—	2.7	—	mA
	V <sub>DD</sub> = 3.0 V, Clock = 25 MHz	—	7	8.9	mA
	V <sub>DD</sub> = 3.0 V, Clock = 1 MHz	—	2.9	—	mA
Digital Supply Current with CPU Inactive (not accessing EPROM)	V <sub>DD</sub> = 1.8 V, Clock = 25 MHz	—	2.2	3	mA
	V <sub>DD</sub> = 1.8 V, Clock = 1 MHz	—	0.41	—	mA
	V <sub>DD</sub> = 3.0 V, Clock = 25 MHz	—	2.3	3.1	mA
	V <sub>DD</sub> = 3.0 V, Clock = 1 MHz	—	0.42	—	mA
Digital Supply Current (shutdown)	Oscillator not running (stop mode), Internal Regulator Off	—	4	—	μA
	Oscillator not running (stop or suspend mode), Internal Regulator On	—	400	—	μA
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
Specified Operating Temperature Range		–40	—	+85	°C
SYSCCLK (system clock frequency)	(Note 2)	0	—	25	MHz
Tsysl (SYSCCLK low time)		18	—	—	ns
Tsysh (SYSCCLK high time)		18	—	—	ns
<b>Notes:</b>					
1. Analog performance is not guaranteed when V <sub>DD</sub> is below 1.8 V.					
2. SYSCCLK must be at least 32 kHz to enable debugging.					

## 8.3.2. 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 8.2 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 "8.3.3. Settling Time Requirements" on page 42.



**Figure 8.2. 10-Bit ADC Track and Conversion Example Timing**

# C8051T610/1/2/3/4/5/6/7

## SFR Definition 12.3. CPT1CN: Comparator1 Control

Bit	7	6	5	4	3	2	1	0
Name	CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP[1:0]		CP1HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9A

Bit	Name	Function
7	CP1EN	<b>Comparator1 Enable Bit.</b> 0: Comparator1 Disabled. 1: Comparator1 Enabled.
6	CP1OUT	<b>Comparator1 Output State Flag.</b> 0: Voltage on CP1+ < CP0-. 1: Voltage on CP1+ > CP0-.
5	CP1RIF	<b>Comparator1 Rising-Edge Flag. Must be cleared by software.</b> 0: No Comparator1 Rising Edge has occurred since this flag was last cleared. 1: Comparator1 Rising Edge has occurred.
4	CP1FIF	<b>Comparator1 Falling-Edge Flag. Must be cleared by software.</b> 0: No Comparator1 Falling-Edge has occurred since this flag was last cleared. 1: Comparator1 Falling-Edge has occurred.
3:2	CP1HYP[1:0]	<b>Comparator1 Positive Hysteresis Control Bits.</b> 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CP1HYN[1:0]	<b>Comparator1 Negative Hysteresis Control Bits.</b> 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

---

## 17. EPROM Memory

Electrically programmable read-only memory (EPROM) is included on-chip for program code storage. The EPROM memory can be programmed via the C2 debug and programming interface when a special programming voltage is applied to the  $V_{PP}$  pin. Each location in EPROM memory is programmable only once (i.e., non-erasable). Table 7.6 on page 34 shows the EPROM specifications.

### 17.1. Programming and Reading the EPROM Memory

Reading and writing the EPROM memory is accomplished through the C2 programming and debug interface. When creating hardware to program the EPROM, it is necessary to follow the programming steps listed below. Refer to the “C2 Interface Specification” available at <http://www.silabs.com> for details on communicating via the C2 interface. Section “27. C2 Interface” on page 208 has information about C2 register addresses for the C8051T610/1/2/3/4/5/6/7.

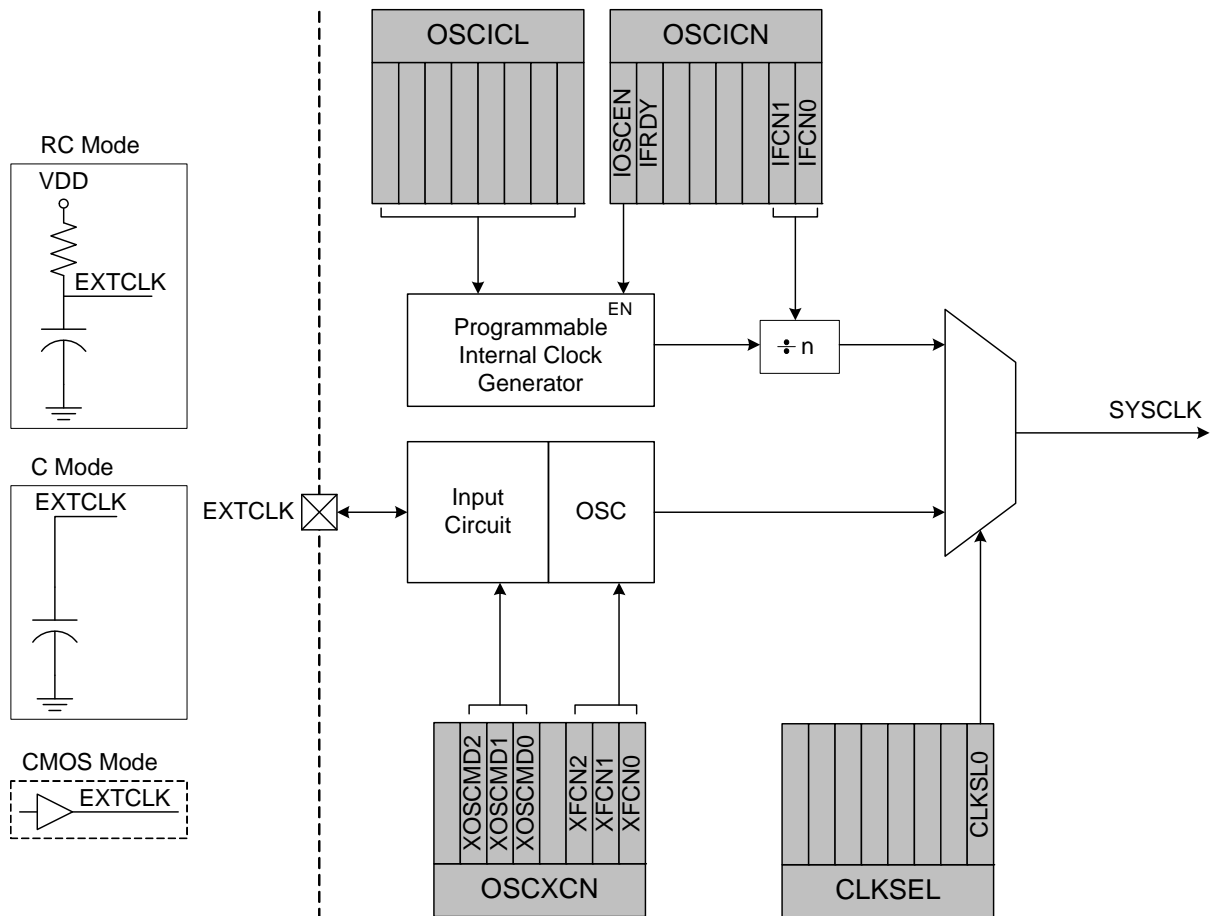
#### 17.1.1. EPROM Write Procedure

1. Reset the device using the  $\overline{RST}$  pin.
2. Wait at least 20  $\mu$ s before sending the first C2 command.
3. Place the device in core reset: Write 0x04 to the DEVCTL register.
4. Set the device to program mode (1st step): Write 0x40 to the EPCTL register.
5. Set the device to program mode (2nd step): Write 0x4A to the EPCTL register.  
**Note:** Prior to date code 1119, 0x58 should be written to EPCTL.
6. Apply the VPP programming Voltage.
7. Write the first EPROM address for programming to EPADDRH and EPADDRL.
8. Write a data byte to EPDAT. EPADDRH:L will increment by 1 after this write.
9. Use a C2 Address Read command to poll for write completion.
10. (Optional) Check the ERROR bit in register EPSTAT and abort the programming operation if necessary.
11. If programming is not finished, return to Step 8 to write the next address in sequence, or return to Step 7 to program a new address.
12. Remove the VPP programming Voltage.
13. Remove program mode (1st step): Write 0x40 to the EPCTL register.
14. Remove program mode (2nd step): Write 0x00 to the EPCTL register.
15. Reset the device: Write 0x02 and then 0x00 to the DEVCTL register.

**Important Note:** There is a finite amount of time which  $V_{PP}$  can be applied without damaging the device, which is cumulative over the life of the device. Refer to Table 7.1 on page 31 for the  $V_{PP}$  timing specification.

## 20. Oscillators and Clock Selection

C8051T610/1/2/3/4/5/6/7 devices include a programmable internal high-frequency oscillator and an external oscillator drive circuit. The internal high-frequency oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 20.1. The system clock can be sourced by the external oscillator circuit or the internal oscillator. The internal oscillator also offers a selectable post-scaling feature.



**Figure 20.1. Oscillator Options**

### 20.1. System Clock Selection

The CLKSL0 bit in register CLKSEL selects which oscillator source is used as the system clock. CLKSL0 must be set to 1 for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals (timers, PCA) when the internal oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal oscillator and external oscillator, so long as the selected clock source is enabled and running.

The internal high-frequency oscillator requires little start-up time and may be selected as the system clock immediately following the register write which enables the oscillator. The external RC and C modes also typically require no startup time.



## 20.2. Programmable Internal High-Frequency (H-F) Oscillator

All C8051T610/1/2/3/4/5/6/7 devices include a programmable internal high-frequency oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register as defined by SFR Definition 20.2.

On C8051T610/1/2/3/4/5/6/7 devices, OSCICL is factory calibrated to obtain a 24.5 MHz base frequency.

The system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN. The divide value defaults to 8 following a reset.

### SFR Definition 20.2. OSCICL: Internal H-F Oscillator Calibration

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

SFR Address = 0xB3

Bit	Name	Function
7	Unused	Unused. Read = 0; Write = Don't Care
6:0	OSCICL[6:0]	<p><b>Internal Oscillator Calibration Bits.</b></p> <p>These bits determine the internal oscillator period. When set to 0000000b, the H-F oscillator operates at its fastest setting. When set to 1111111b, the H-F oscillator operates at its slowest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.</p>

# C8051T610/1/2/3/4/5/6/7

## 21.5. Special Function Registers for Accessing and Configuring Port I/O

All Port I/O are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable. When writing to a Port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the Port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the Crossbar, the Port register can always read its corresponding Port I/O pin). The exception to this is the execution of the read-modify-write instructions that target a Port Latch register as the destination. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a Port SFR. For these instructions, the value of the latch register (not the pin) is read, modified, and written back to the SFR.

Each Port has a corresponding PnSKIP register which allows its individual Port pins to be assigned to digital functions or skipped by the Crossbar. All Port pins used for analog functions, GPIO, or dedicated digital functions such as the EMIF should have their PnSKIP bit set to 1.

The Port input mode of the I/O pins is defined using the Port Input Mode registers (PnMDIN). Each Port cell can be configured for analog or digital I/O. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings.

### SFR Definition 21.3. P0: Port 0

Bit	7	6	5	4	3	2	1	0
Name	P0[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x80; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P0[7:0]	<b>Port 0 Data.</b> Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P0.n Port pin is logic LOW. 1: P0.n Port pin is logic HIGH.

# C8051T610/1/2/3/4/5/6/7

## SFR Definition 21.10. P1SKIP: Port 1 Skip

Bit	7	6	5	4	3	2	1	0
Name	P1SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD5

Bit	Name	Function
7:0	P1SKIP[7:0]	<p><b>Port 1 Crossbar Skip Enable Bits.</b></p> <p>These bits select Port 1 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar.</p> <p>0: Corresponding P1.n pin is not skipped by the Crossbar.</p> <p>1: Corresponding P1.n pin is skipped by the Crossbar.</p>
<p><b>Note:</b> P1.6 and P1.7 are not connected to external pins on the C8051T616/7 devices. When writing code for the C8051T616/7, P1SKIP[6:7] should be set to 11b to skip these two pins on the crossbar.</p>		

## SFR Definition 21.11. P2: Port 2

Bit	7	6	5	4	3	2	1	0
Name	P2[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xA0; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P2[7:0]	<p><b>Port 2 Data.</b></p> <p>Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.</p>	<p>0: Set output latch to logic LOW.</p> <p>1: Set output latch to logic HIGH.</p>	<p>0: P2.n Port pin is logic LOW.</p> <p>1: P2.n Port pin is logic HIGH.</p>
<p><b>Note:</b> P2.6 and P2.7 are not connected to external pins on the C8051T616/7 devices.</p>				

# C8051T610/1/2/3/4/5/6/7

## 22.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I<sup>2</sup>C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I<sup>2</sup>C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

## 22.2. SMBus Configuration

Figure 22.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

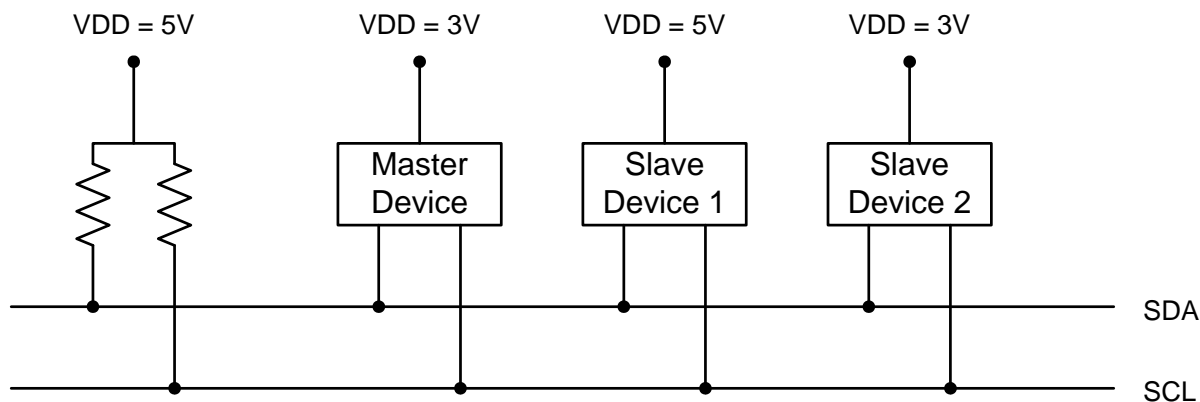


Figure 22.2. Typical SMBus Configuration

## 22.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. Note that it is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 22.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

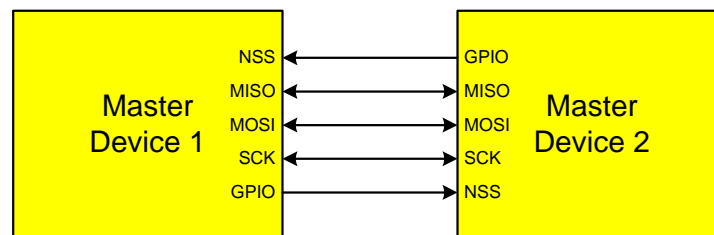
## 24.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 24.2 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 24.3 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 24.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.



**Figure 24.2. Multiple-Master Mode Connection Diagram**

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 24.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

## 24.4. 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.

## 24.5. Serial Clock Phase and Polarity

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 24.5. For slave mode, the clock and data relationships are shown in Figure 24.6 and Figure 24.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 24.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

## 25.1. Timer 0 and Timer 1

Each timer is implemented as a 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 (Section “16.2. Interrupt Register Descriptions” on page 87); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section “16.2. Interrupt Register Descriptions” on page 87). 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.

### 25.1.1. 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 in TCON is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit in the TMOD register 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 “21.3. Priority Crossbar Decoder” on page 117 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit in register CKCON. 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 25.1).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or the input signal  $\overline{INT0}$  is active as defined by bit IN0PL in register IT01CF (see SFR Definition 16.5). Setting GATE0 to 1 allows the timer to be controlled by the external input signal  $\overline{INT0}$  (see Section “16.2. Interrupt Register Descriptions” on page 87), facilitating pulse width measurements

TR0	GATE0	$\overline{INT0}$	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

**Note:** 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  $\overline{INT0}$  is used with Timer 1; the  $\overline{INT1}$  polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 16.5).

# C8051T610/1/2/3/4/5/6/7

## 25.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

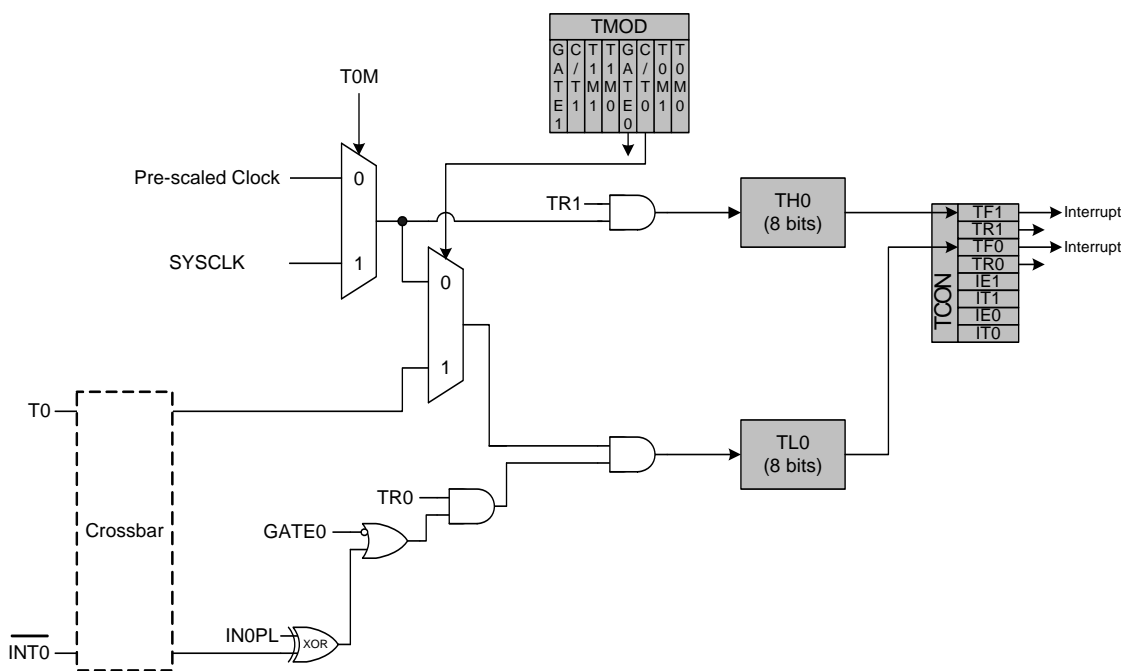


Figure 25.3. T0 Mode 3 Block Diagram



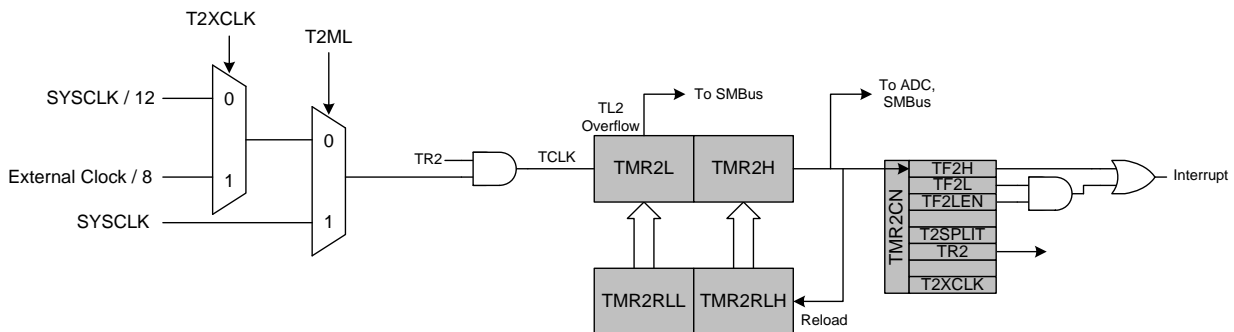
## 25.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. 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 T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by 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 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 25.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled, an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.



**Figure 25.4. Timer 2 16-Bit Mode Block Diagram**



## 26.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 4. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH4) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

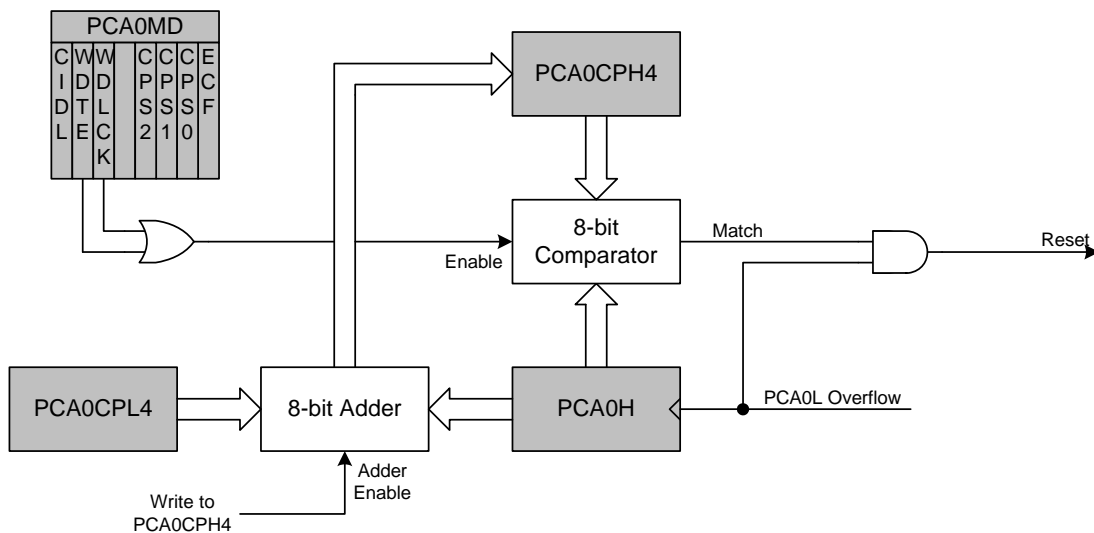
With the WDTE bit set in the PCA0MD register, Module 4 operates as a watchdog timer (WDT). The Module 4 high byte is compared to the PCA counter high byte; the Module 4 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

### 26.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2–CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 4 is forced into software timer mode.
- Writes to the Module 4 mode register (PCA0CPM4) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH4 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH4. Upon a PCA0CPH4 write, PCA0H plus the offset held in PCA0CPL4 is loaded into PCA0CPH4 (See Figure 26.10).



**Figure 26.10. PCA Module 4 with Watchdog Timer Enabled**

## SFR Definition 26.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
<b>Name</b>	CIDL	WDTE	WDLCK			CPS[2:0]		ECF
<b>Type</b>	R/W	R/W	R/W	R		R/W		R/W
<b>Reset</b>	0	1	0	0	0	0	0	0

SFR Address = 0xD9

Bit	Name	Function
7	CIDL	<p><b>PCA Counter/Timer Idle Control.</b>                      Specifies PCA behavior when CPU is in Idle Mode.                      0: PCA continues to function normally while the system controller is in Idle Mode.                      1: PCA operation is suspended while the system controller is in Idle Mode.</p>
6	WDTE	<p><b>Watchdog Timer Enable.</b>                      If this bit is set, PCA Module 4 is used as the watchdog timer.                      0: Watchdog Timer disabled.                      1: PCA Module 4 enabled as Watchdog Timer.</p>
5	WDLCK	<p><b>Watchdog Timer Lock.</b>                      This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset.                      0: Watchdog Timer Enable unlocked.                      1: Watchdog Timer Enable locked.</p>
4	Unused	Unused. Read = 0b, Write = Don't care.
3:1	CPS[2:0]	<p><b>PCA Counter/Timer Pulse Select.</b>                      These bits select the timebase source for the PCA counter                      000: System clock divided by 12                      001: System clock divided by 4                      010: Timer 0 overflow                      011: High-to-low transitions on ECI (max rate = system clock divided by 4)                      100: System clock                      101: External clock divided by 8 (synchronized with the system clock)                      11x: Reserved</p>
0	ECF	<p><b>PCA Counter/Timer Overflow Interrupt Enable.</b>                      This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt.                      0: Disable the CF interrupt.                      1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.</p>

**Note:** When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.

# C8051T610/1/2/3/4/5/6/7

## SFR Definition 26.6. PCA0CPLn: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[7:0]							
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: 0xFB (n = 0), 0xE9 (n = 1), 0xEB (n = 2), 0xED (n = 3), 0xFD (n = 4)

Bit	Name	Function
7:0	PCA0CPn[7:0]	<b>PCA Capture Module Low Byte.</b> The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.
<b>Note:</b> A write to this register will clear the module's ECOMn bit to a 0.		

## SFR Definition 26.7. PCA0CPHn: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[15:8]							
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: 0xFC (n = 0), 0xEA (n = 1), 0xEC (n = 2), 0xEE (n = 3), 0xFE (n = 4)

Bit	Name	Function
7:0	PCA0CPn[15:8]	<b>PCA Capture Module High Byte.</b> The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.
<b>Note:</b> A write to this register will set the module's ECOMn bit to a 1.		