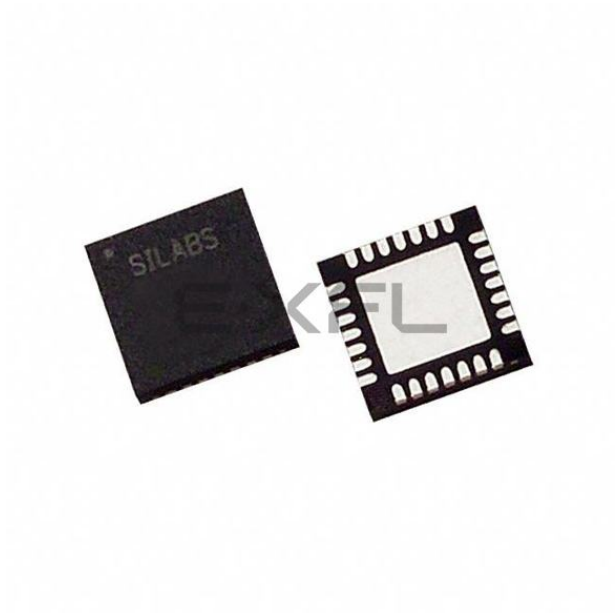



**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 25 |
| 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 17x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-VFQFN Exposed Pad |
| Supplier Device Package | 28-QFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051t611-gm |

## 6. QFN-24 Package Specifications

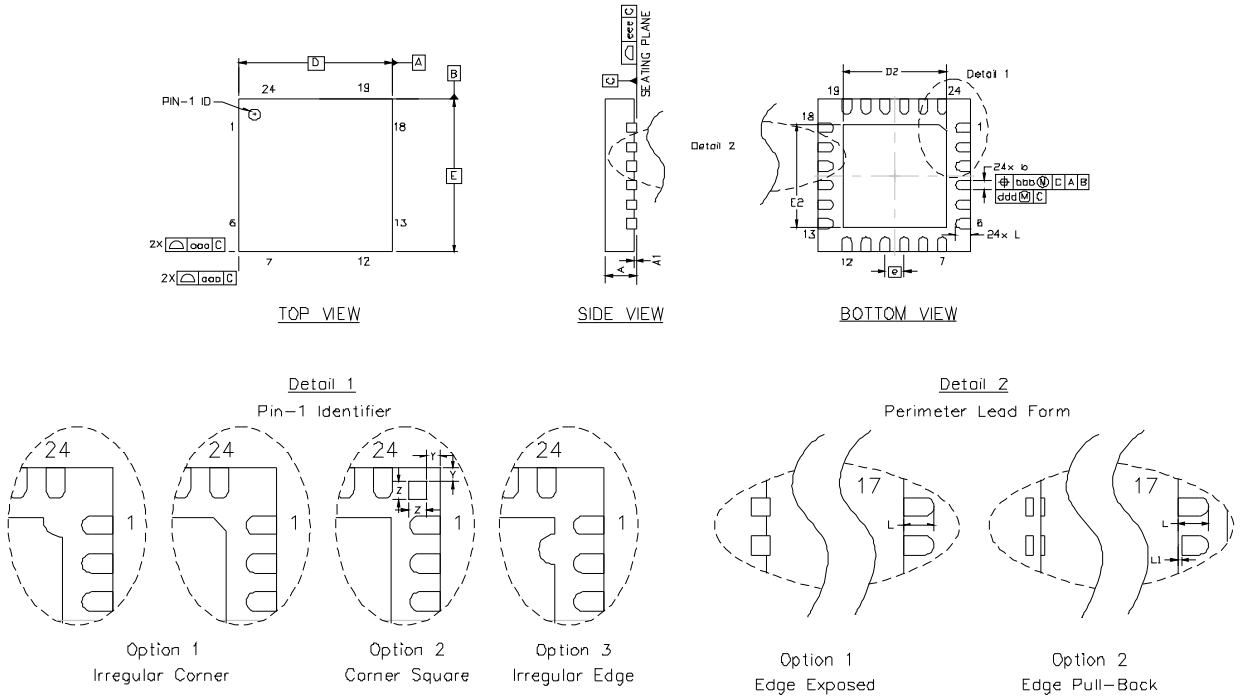



**Figure 6.1. QFN-24 Package Drawing**

**Table 6.1. QFN-24 Package Dimensions**

| Dimension | Min | Typ | Max | | Dimension | Min | Typ | Max |
|---|---|---|---|---|---|---|---|---|
| A | 0.70 | 0.75 | 0.80 | | L | 0.30 | 0.40 | 0.50 |
| A1 | 0.00 | 0.02 | 0.05 | | L1 | 0.00 | — | 0.15 |
| b | 0.18 | 0.25 | 0.30 | | aaa | — | — | 0.15 |
| D | 4.00 BSC. | | | | bbb | — | — | 0.10 |
| D2 | 2.55 | 2.70 | 2.80 | | ddd | — | — | 0.05 |
| e | 0.50 BSC. | | | | eee | — | — | 0.08 |
| E | 4.00 BSC. | | | | Z | — | 0.24 | — |
| E2 | 2.55 | 2.70 | 2.80 | | Y | — | 0.18 | — |

**Notes:**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation WGGD except for custom features D2, E2, Z, Y, and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

### 8.3.3. Settling Time Requirements

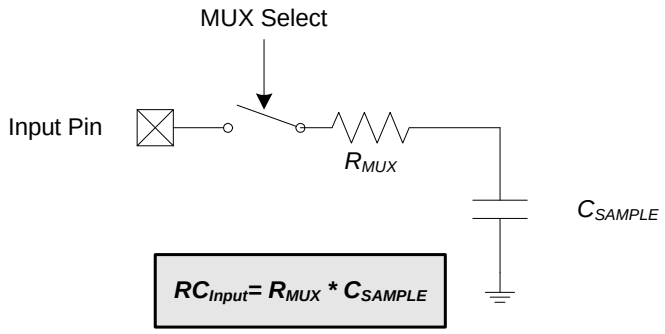
A minimum tracking time is required before each conversion to ensure that an accurate conversion is performed. This tracking time is determined by any series impedance, including the AMUX0 resistance, the the ADC0 sampling capacitance, and the accuracy required for the conversion. Note that in delayed tracking mode, three SAR clocks are used for tracking at the start of every conversion. For many applications, these three SAR clocks will meet the minimum tracking time requirements.

Figure 8.3 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 8.1. See Table 7.8 for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

**Equation 8.1. ADC0 Settling Time Requirements**

Where:
*SA* is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
*t* is the required settling time in seconds
$R_{TOTAL}$ is the sum of the AMUX0 resistance and any external source resistance.
*n* is the ADC resolution in bits (10).




Note: See electrical specification tables for $R_{MUX}$ and $C_{SAMPLE}$ parameters.

**Figure 8.3. ADC0 Equivalent Input Circuits**

### 8.4.1. Window Detector Example

Figure 8.4 shows two example window comparisons for right-justified data, with ADC0LTH:ADC0LTL = 0x0080 (128d) and ADC0GTH:ADC0GTL = 0x0040 (64d). The input voltage can range from 0 to VREF x (1023/1024) with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0040 < ADC0H:ADC0L < 0x0080). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0040 or ADC0H:ADC0L > 0x0080). Figure 8.5 shows an example using left-justified data with the same comparison values.

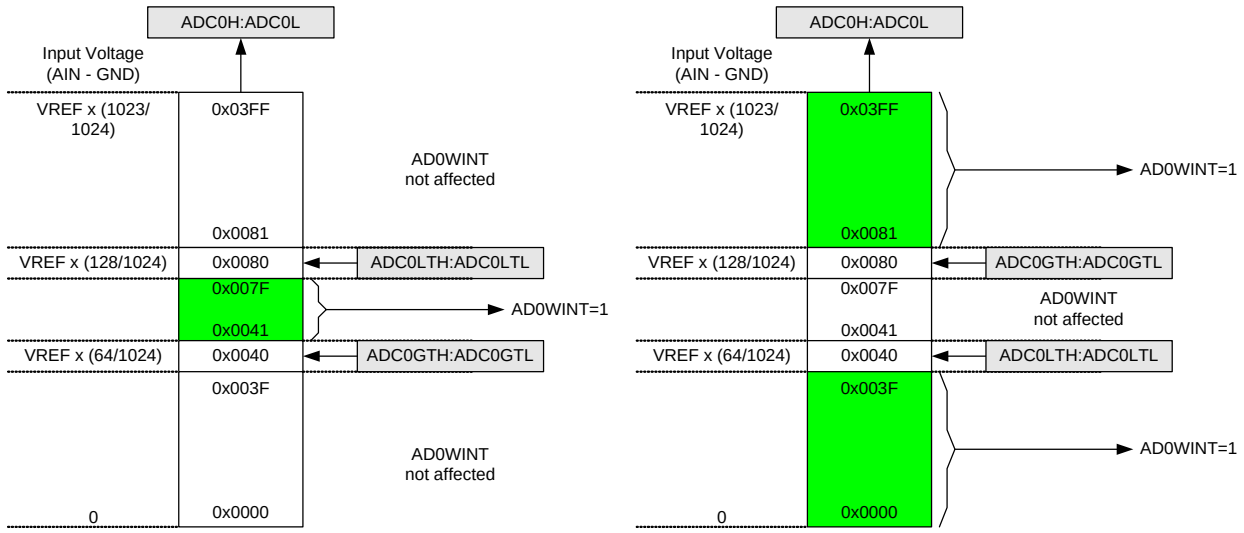



**Figure 8.4. ADC Window Compare Example: Right-Justified Data**

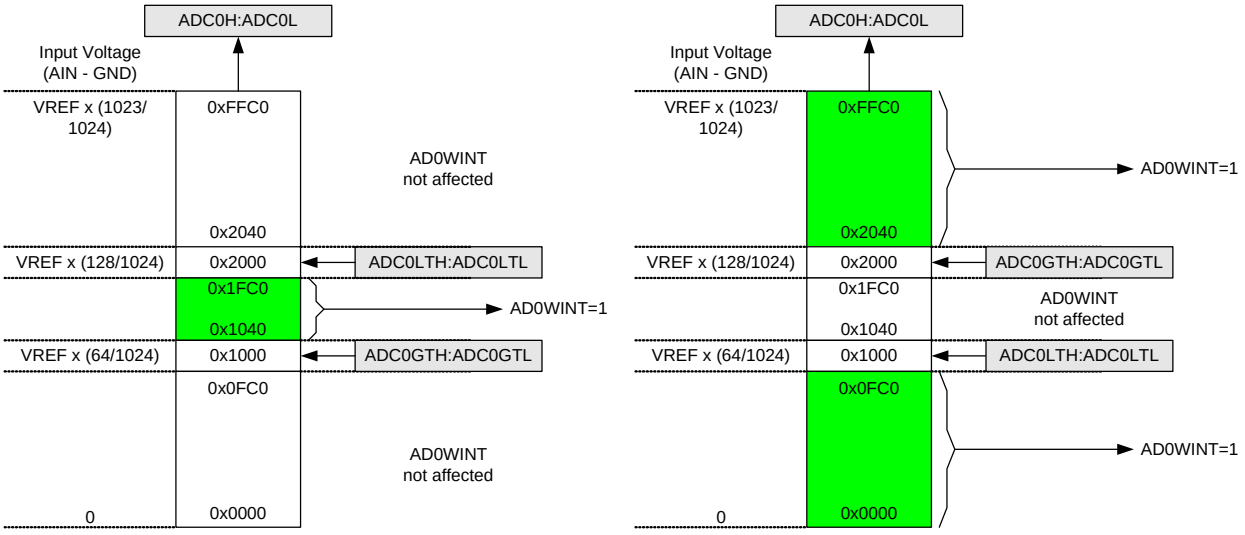



**Figure 8.5. ADC Window Compare Example: Left-Justified Data**

## 8.5. ADC0 Analog Multiplexer (C8051T610/1/2/3/6 only)

ADC0 on the C8051T610/1/2/3/6 uses an analog input multiplexer to select the positive input to the ADC. Any of the following may be selected as the positive input: Port 1, 2 and 3 I/O pins, the on-chip temperature sensor, or the positive power supply ($V_{DD}$). The ADC0 input channel is selected in the AMX0P register described in SFR Definition 8.9.

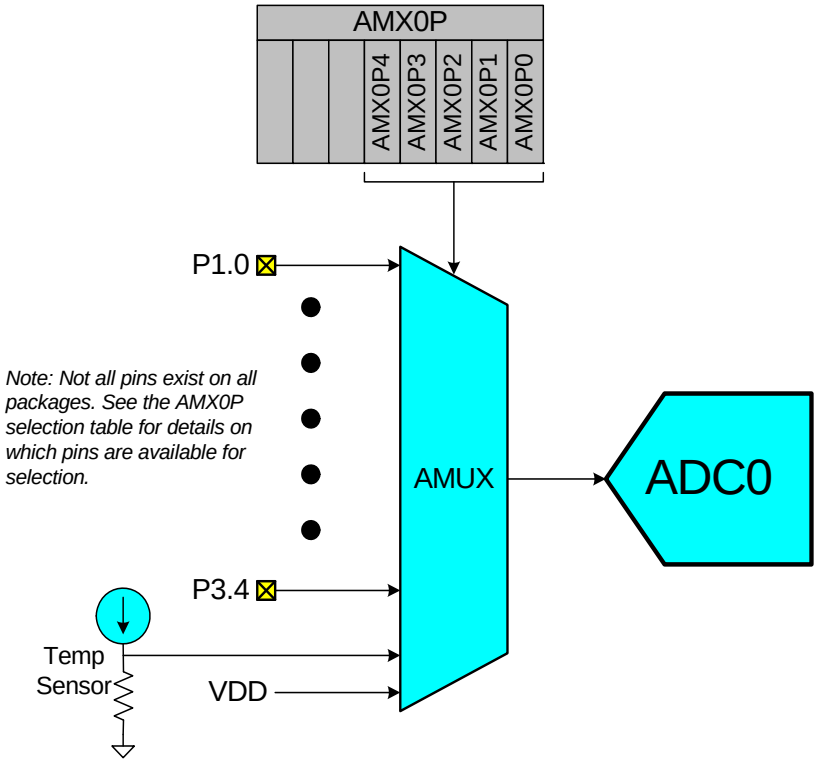



**Figure 8.6. ADC0 Multiplexer Block Diagram**

**Important Note About ADC0 Input Configuration:** Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN. To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section “21. Port Input/Output” on page 113 for more Port I/O configuration details.

## SFR Definition 8.9. AMX0P: AMUX0 Positive Channel Select

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | AMX0P[4:0] | | | | |
| Type | R | R | R | R/W | | | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

SFR Address = 0xBB

| Bit | Name | Function | | |
|---|---|---|---|---|
| 7:5 | Unused | Unused. Read = 000b; Write = Don't Care. | | |
| 4:0 | AMX0P[4:0] | **AMUX0 Positive Input Selection.** | | |
| | | Setting | Channel | Available on Packages |
| | | 00000: | P1.0 | LQFP-32, QFN-28, QFN-24 |
| | | 00001: | P1.1 | LQFP-32, QFN-28, QFN-24 |
| | | 00010: | P1.2 | LQFP-32, QFN-28, QFN-24 |
| | | 00011: | P1.3 | LQFP-32, QFN-28, QFN-24 |
| | | 00100: | P1.4 | LQFP-32, QFN-28, QFN-24 |
| | | 00101: | P1.5 | LQFP-32, QFN-28, QFN-24 |
| | | 00110: | P1.6 | LQFP-32, QFN-28 |
| | | 00111: | P1.7 | LQFP-32, QFN-28 |
| | | 01000: | P2.0 | LQFP-32, QFN-28, QFN-24 |
| | | 01001: | P2.1 | LQFP-32, QFN-28, QFN-24 |
| | | 01010: | P2.2 | LQFP-32, QFN-28, QFN-24 |
| | | 01011: | P2.3 | LQFP-32, QFN-28, QFN-24 |
| | | 01100: | P2.4 | LQFP-32, QFN-28, QFN-24 |
| | | 01101: | P2.5 | LQFP-32, QFN-28, QFN-24 |
| | | 01110: | P2.6 | LQFP-32, QFN-28 |
| | | 01111: | P2.7 | LQFP-32, QFN-28 |
| | | 10000: | P3.0 | LQFP-32, QFN-28, QFN-24 |
| | | 10001: | P3.1 | LQFP-32 |
| | | 10010: | P3.2 | LQFP-32 |
| | | 10011: | P3.3 | LQFP-32 |
| | | 10100: | P3.4 | LQFP-32 |
| | | 10101-11101: | No Input Selected | N/A |
| | | 11110: | Temp Sensor | LQFP-32, QFN-28, QFN-24 |
| | | 11111: | $V_{DD}$ | LQFP-32, QFN-28, QFN-24 |

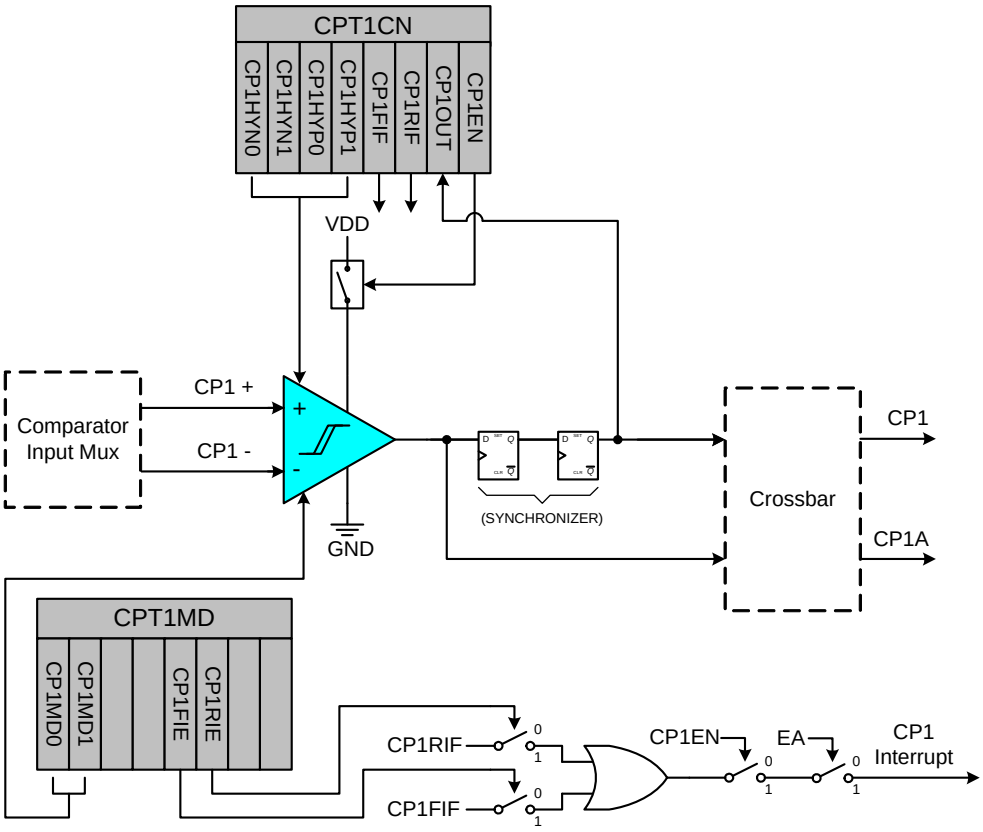



**Figure 12.2. Comparator1 Functional Block Diagram**

The Comparator output can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and the power supply to the comparator is turned off. See Section “21.3. Priority Crossbar Decoder” on page 117 for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from –0.25 V to ($V_{DD}$) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Section “7. Electrical Characteristics” on page 31.

The Comparator response time may be configured in software via the CPTnMD registers (see SFR Definition 12.2 and SFR Definition 12.4). Selecting a longer response time reduces the Comparator supply current.

## SFR Definition 12.1. CPT0CN: Comparator0 Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CP0EN | CP0OUT | CP0RIF | CP0FIF | CP0HYP[1:0] | | CP0HYN[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 = 0x9B

| Bit | Name | Function |
|---|---|---|
| 7 | CP0EN | **Comparator0 Enable Bit.**<br>0: Comparator0 Disabled.<br>1: Comparator0 Enabled. |
| 6 | CP0OUT | **Comparator0 Output State Flag.**<br>0: Voltage on CP0+ < CP0–.<br>1: Voltage on CP0+ > CP0–. |
| 5 | CP0RIF | **Comparator0 Rising-Edge Flag. Must be cleared by software.**<br>0: No Comparator0 Rising Edge has occurred since this flag was last cleared.<br>1: Comparator0 Rising Edge has occurred. |
| 4 | CP0FIF | **Comparator0 Falling-Edge Flag. Must be cleared by software.**<br>0: No Comparator0 Falling-Edge has occurred since this flag was last cleared.<br>1: Comparator0 Falling-Edge has occurred. |
| 3:2 | CP0HYP[1:0] | **Comparator0 Positive Hysteresis Control Bits.**<br>00: Positive Hysteresis Disabled.<br>01: Positive Hysteresis = 5 mV.<br>10: Positive Hysteresis = 10 mV.<br>11: Positive Hysteresis = 20 mV. |
| 1:0 | CP0HYN[1:0] | **Comparator0 Negative Hysteresis Control Bits.**<br>00: Negative Hysteresis Disabled.<br>01: Negative Hysteresis = 5 mV.<br>10: Negative Hysteresis = 10 mV.<br>11: Negative Hysteresis = 20 mV. |

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. 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 |

## 13.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

### 13.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 13.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

#### 14.2.1.1. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 13.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

#### 14.2.1.2. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

#### 14.2.1.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

### 14.2.2. External RAM

There are 1024 bytes of on-chip RAM mapped into the external data memory space. All of these address locations may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN as shown in SFR Definition 14.1).

For a 16-bit MOVX operation (@DPTR), the upper 7 bits of the 16-bit external data memory address word are "don't cares". As a result, the 1024-byte RAM is mapped modulo style over the entire 64 k external data memory address range. For example, the XRAM byte at address 0x0000 is shadowed at addresses 0x0400, 0x0800, 0x0C00, 0x1000, etc. This is a useful feature when performing a linear memory fill, as the address pointer doesn't have to be reset when reaching the RAM block boundary.

## 19. Reset Sources

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

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For $V_{DD}$ Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source. Program execution begins at location 0x0000.

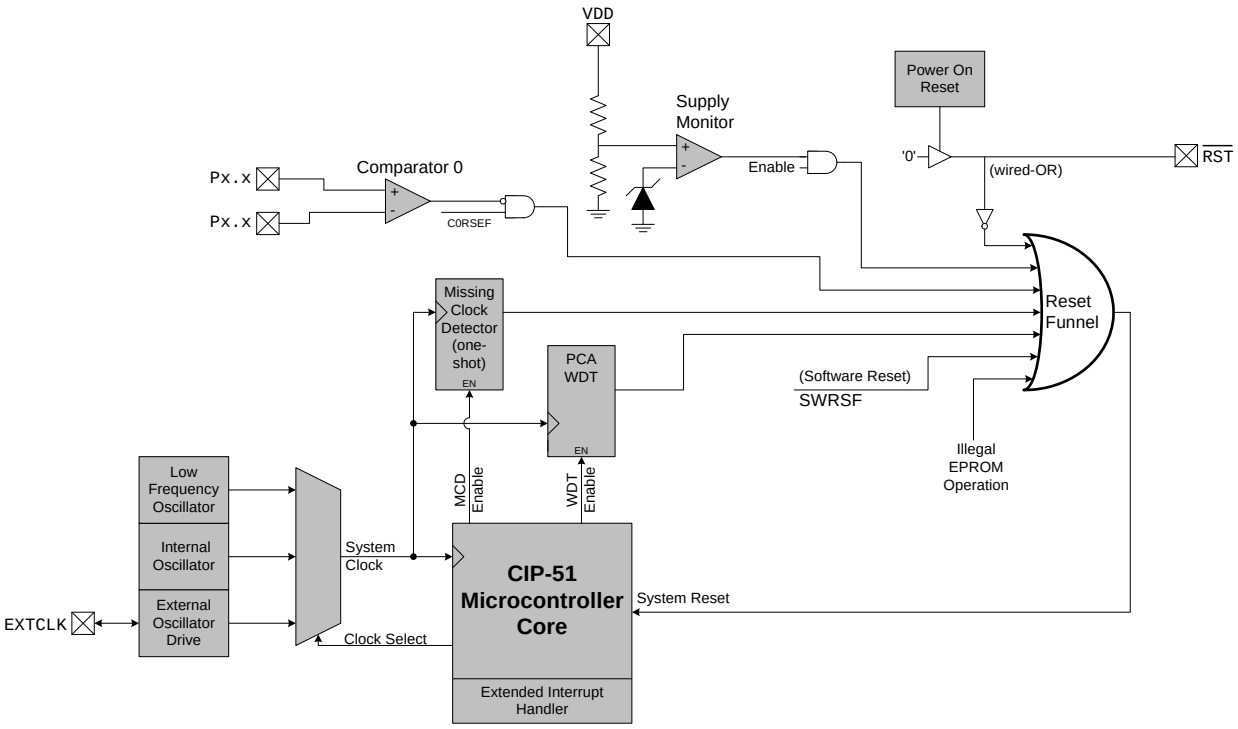



**Figure 19.1. Reset Sources**

# 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.

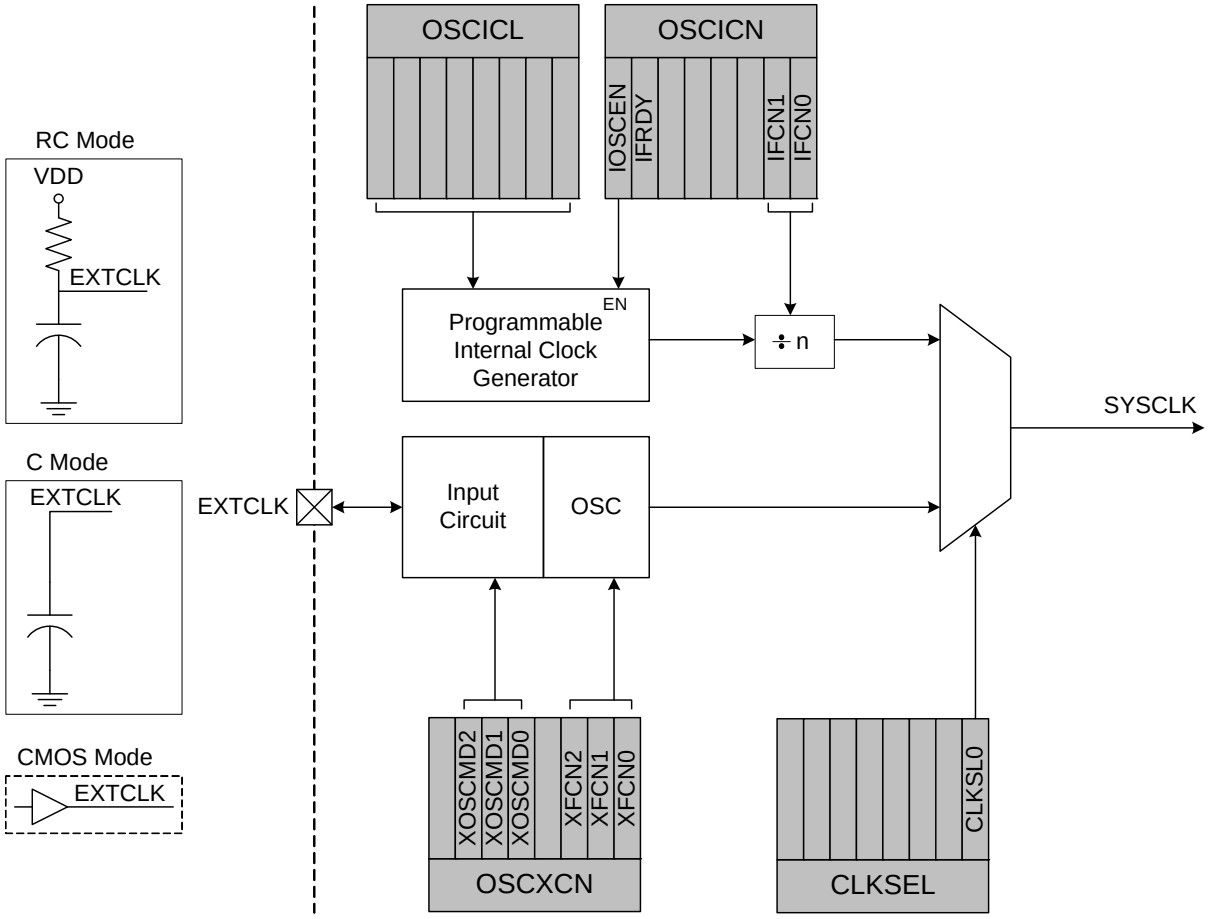



**Figure 20.1. Oscillator Options**

## 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.

# 21. Port Input/Output

Digital and analog resources are available through 29 I/O pins organized as three byte-wide ports and one 5-bit-wide port on the C8051T610/2/4. The C8051T611/3/5 devices have 25 I/O pins available, organized as three byte-wide ports and one 1-bit-wide port. The C8051T616/7 have 21 I/O pins available on a single byte-wide port, two 6-bit-wide ports, and a 1-bit-wide port.

Port pins can be defined as general-purpose I/O (GPIO), assigned to one of the internal digital resources, or assigned to an analog function as shown in Figure 21.3. Port pin P3.0 is shared with the C2 Interface Data signal (C2D). 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. Note that 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 21.3, Figure 21.4, and Figure 21.5). The registers XBR0 and XBR1, defined in SFR Definition 21.1 and SFR Definition 21.2, are used to select internal digital functions.

All Port I/O pins are 5 V tolerant (refer to Figure 21.2 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 7.3 on page 33.

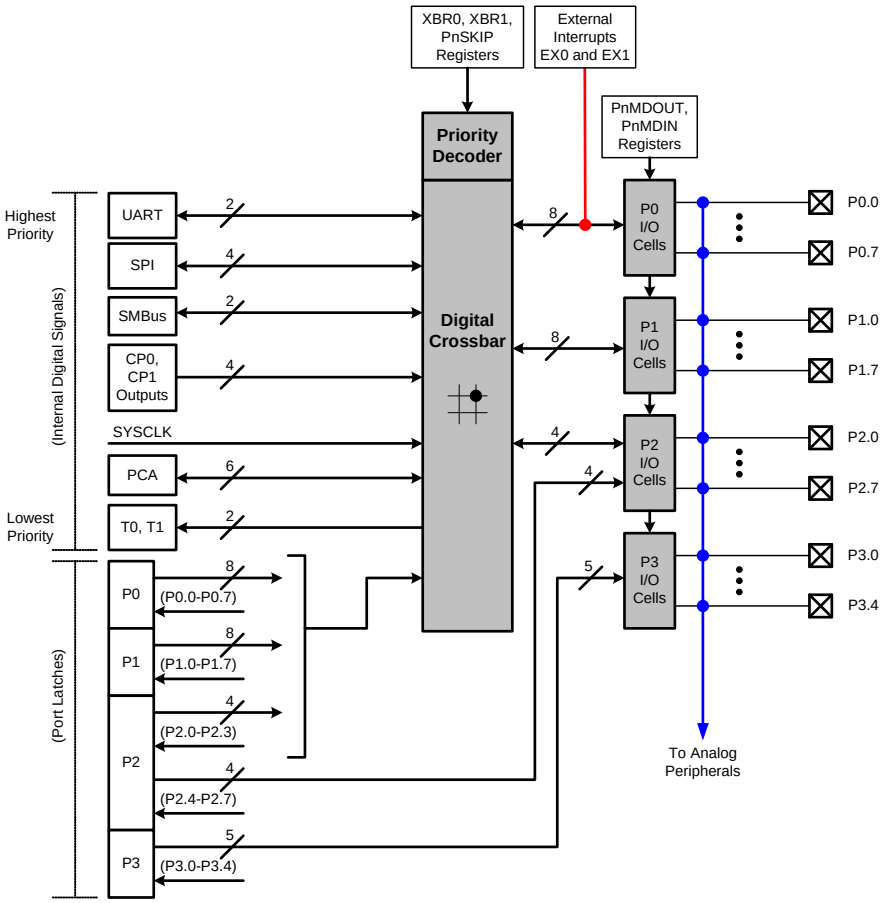



**Figure 21.1. Port I/O Functional Block Diagram**

## SFR Definition 24.2. SPI0CN: SPI0 Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SPIF | WCOL | MODF | RXOVRN | NSSMD[1:0] | | TXBMT | SPIEN |
| Type | R/W | R/W | R/W | R/W | R/W | | R | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

SFR Address = 0xF8; Bit-Addressable

| Bit | Name | Function |
|---|---|---|
| 7 | SPIF | **SPI0 Interrupt Flag.**<br>This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software. |
| 6 | WCOL | **Write Collision Flag.**<br>This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software. |
| 5 | MODF | **Mode Fault Flag.**<br>This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software. |
| 4 | RXOVRN | **Receive Overrun Flag (valid in slave mode only).**<br>This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software. |
| 3:2 | NSSMD[1:0] | **Slave Select Mode.**<br>Selects between the following NSS operation modes:<br>(See Section 24.2 and Section 24.3).<br>00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin.<br>01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device.<br>1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0. |
| 1 | TXBMT | **Transmit Buffer Empty.**<br>This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer. |
| 0 | SPIEN | **SPI0 Enable.**<br>0: SPI disabled.<br>1: SPI enabled. |

**Table 24.1. SPI Slave Timing Parameters**

| Parameter | Description | Min | Max | Units |
|---|---|---|---|---|
| **Master Mode Timing** (See Figure 24.8 and Figure 24.9) | | | | |
| $T_{MCKH}$ | SCK High Time | 1 x $T_{SYSCLK}$ | — | ns |
| $T_{MCKL}$ | SCK Low Time | 1 x $T_{SYSCLK}$ | — | ns |
| $T_{MIS}$ | MISO Valid to SCK Shift Edge | 1 x $T_{SYSCLK}$ + 20 | — | ns |
| $T_{MIH}$ | SCK Shift Edge to MISO Change | 0 | — | ns |
| **Slave Mode Timing** (See Figure 24.10 and Figure 24.11) | | | | |
| $T_{SE}$ | NSS Falling to First SCK Edge | 2 x $T_{SYSCLK}$ | — | ns |
| $T_{SD}$ | Last SCK Edge to NSS Rising | 2 x $T_{SYSCLK}$ | — | ns |
| $T_{SEZ}$ | NSS Falling to MISO Valid | — | 4 x $T_{SYSCLK}$ | ns |
| $T_{SDZ}$ | NSS Rising to MISO High-Z | — | 4 x $T_{SYSCLK}$ | ns |
| $T_{CKH}$ | SCK High Time | 5 x $T_{SYSCLK}$ | — | ns |
| $T_{CKL}$ | SCK Low Time | 5 x $T_{SYSCLK}$ | — | ns |
| $T_{SIS}$ | MOSI Valid to SCK Sample Edge | 2 x $T_{SYSCLK}$ | — | ns |
| $T_{SIH}$ | SCK Sample Edge to MOSI Change | 2 x $T_{SYSCLK}$ | — | ns |
| $T_{SOH}$ | SCK Shift Edge to MISO Change | — | 4 x $T_{SYSCLK}$ | ns |
| $T_{SLH}$ | Last SCK Edge to MISO Change (CKPHA = 1 ONLY) | 6 x $T_{SYSCLK}$ | 8 x $T_{SYSCLK}$ | ns |
| **Note:** $T_{SYSCLK}$ is equal to one period of the device system clock (SYSCLK). | | | | |

### 25.3.2. 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 25.7. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for 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 |

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.

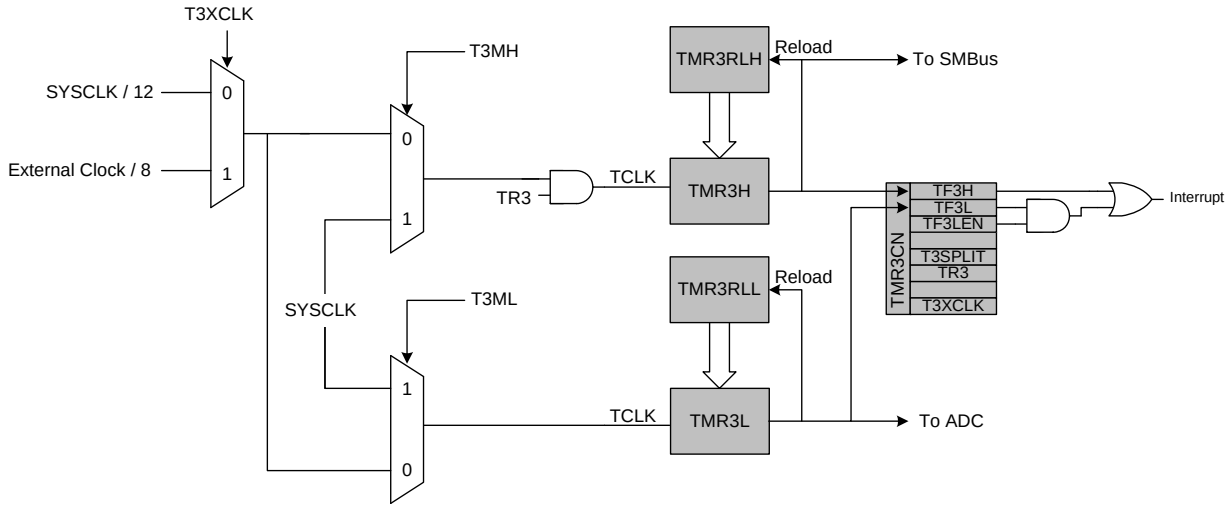



**Figure 25.7. Timer 3 8-Bit Mode Block Diagram**

## SFR Definition 25.14. TMR3RLL: Timer 3 Reload Register Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3RLL[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0x92

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3RLL[7:0] | **Timer 3 Reload Register Low Byte.**<br>TMR3RLL holds the low byte of the reload value for Timer 3. |

## SFR Definition 25.15. TMR3RLH: Timer 3 Reload Register High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3RLH[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0x93

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3RLH[7:0] | **Timer 3 Reload Register High Byte.**<br>TMR3RLH holds the high byte of the reload value for Timer 3. |

## SFR Definition 25.16. 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

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3L[7:0] | **Timer 3 Low Byte.**<br>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. |

### 26.3.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode. If ECOMn is cleared, the associated pin will retain its state, and not toggle on the next match event.

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

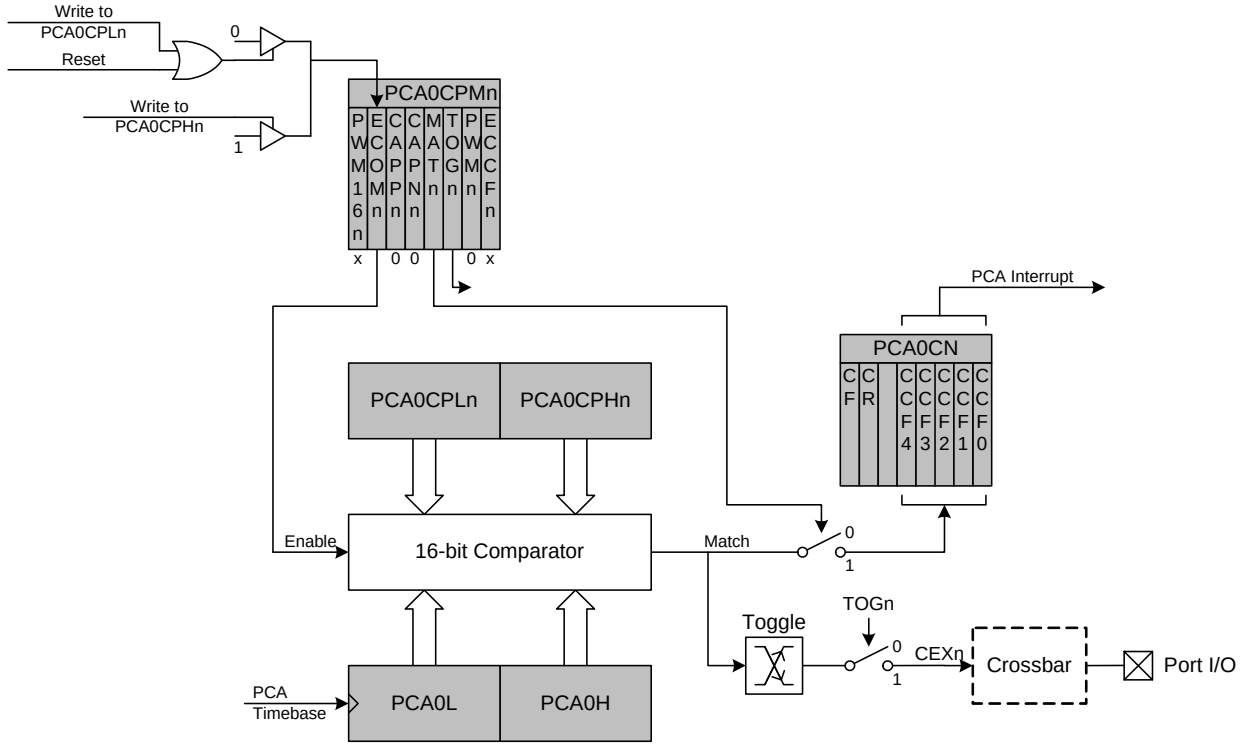



**Figure 26.6. PCA High-Speed Output Mode Diagram**

## 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] | **PCA Capture Module Low Byte.**<br>The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. |
| **Note:** 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] | **PCA Capture Module High Byte.**<br>The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. |
| **Note:** A write to this register will set the module's ECOMn bit to a 1. | | |

## C2 Register Definition 27.4. DEVCTL: C2 Device Control

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

C2 Address: 0x02

| Bit | Name | Function |
|---|---|---|
| 7:0 | DEVCTL[7:0] | **Device Control Register.**<br>This register is used to halt the device for EPROM operations via the C2 interface. Refer to the EPROM chapter for more information. |

## C2 Register Definition 27.5. EPCTL: EPROM Programming Control Register

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

C2 Address: 0xDF

| Bit | Name | Function |
|---|---|---|
| 7:0 | EPCTL[7:0] | **EPROM Programming Control Register.**<br>This register is used to enable EPROM programming via the C2 interface. Refer to the EPROM chapter for more information. |

## C2 Register Definition 27.8. EPADDRH: C2 EPROM Address High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | EPADDR[15:8] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C2 Address: 0xAF

| Bit | Name | Function |
|---|---|---|
| 7:0 | EPADDR[15:8] | **C2 EPROM Address High Byte.**<br>This register is used to set the EPROM address location during C2 EPROM operations. |

## C2 Register Definition 27.9. EPADDRL: C2 EPROM Address Low Byte

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

C2 Address: 0xAE

| Bit | Name | Function |
|---|---|---|
| 7:0 | EPADDR[15:8] | **C2 EPROM Address Low Byte.**<br>This register is used to set the EPROM address location during C2 EPROM operations. |