**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 | Last Time Buy |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | SMBus (2-Wire/I²C), UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 8 |
| Program Memory Size | 8KB (8K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 8x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 14-SOIC (0.154", 3.90mm Width) |
| Supplier Device Package | 14-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f302-gs |

## List of Registers

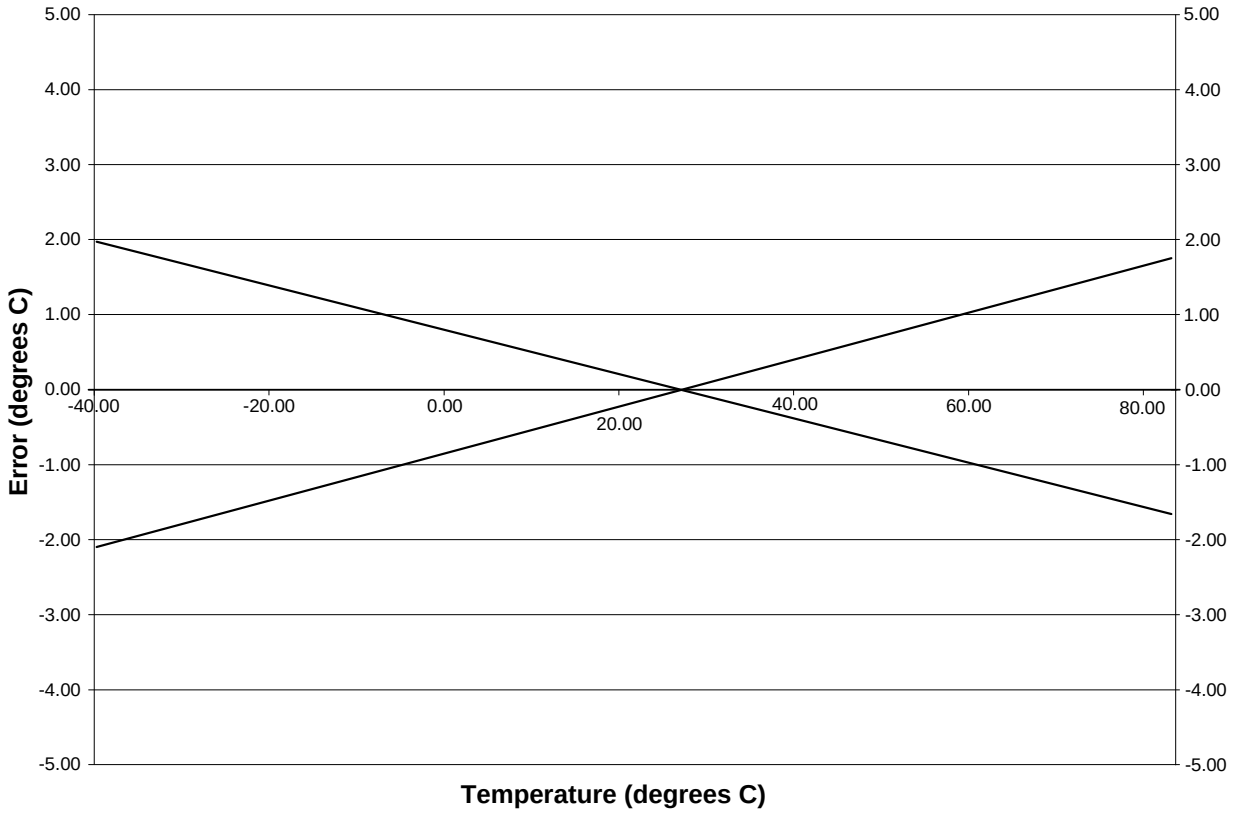


**Figure 5.3. Temperature Sensor Error with 1-Point Calibration (VREF = 2.40 V)**

## Table 5.1. ADC0 Electrical Characteristics

$V_{DD}$ = 3.0 V, VREF = 2.40 V (REFSL = 0), PGA Gain = 1, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 8 | | bits |
| Integral Nonlinearity | | — | ±0.5 | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | ±0.5 | ±1 | LSB |
| Offset Error | | –5.0 | 0.5 | 5.0 | LSB |
| Full Scale Error | Differential mode | –5.0 | –1 | 5.0 | LSB |
| **Dynamic Performance (10 kHz Sine-wave Differential Input, 1 dB below Full Scale, 500 ksps)** | | | | | |
| Signal-to-Noise Plus Distortion | | 45 | 48 | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | –56 | — | dB |
| Spurious-Free Dynamic Range | | — | 58 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Conversion Clock | | — | — | 6 | MHz |
| Conversion Time in SAR Clocks | | 11 | — | — | clocks |
| Track/Hold Acquisition Time | | 300 | — | — | ns |
| Throughput Rate | | — | — | 500 | ksps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | | 0 | — | VREF | V |
| Input Capacitance | | — | 5 | — | pF |
| **Temperature Sensor** | | — | — | — | |
| Linearity[1,2,3] | | — | ±0.5 | — | °C |
| Gain[1,2,3] | | — | 3350 ±110 | — | µV / °C |
| Offset[1,2,3] | (Temp = 0 °C) | — | 897±31 | — | mV |
| **Power Specifications** | | | | | |
| Power Supply Current ($V_{DD}$ supplied to ADC0) | Operating Mode, 500 ksps | — | 400 | 900 | µA |
| Power Supply Rejection | | — | ±0.3 | — | mV/V |

**Notes:**
1. Represents one standard deviation from the mean.
2. Measured with PGA Gain = 2.
3. Includes ADC offset, gain, and linearity variations.

The output of Comparator0 can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator0 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 Comparator0 output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to less than 100 nA. See **Section "12.1. Priority Crossbar Decoder" on page 104** for details on configuring the Comparator0 output via the digital Crossbar. Comparator0 inputs can be externally driven from –0.25 to ($V_{DD}$) + 0.25 V without damage or upset. The complete electrical specifications for Comparator0 are given in Table 7.1.

The Comparator0 response time may be configured in software via the CP0MD1-0 bits in register CPT0MD (see SFR Definition 7.3). Selecting a longer response time reduces the amount of power consumed by Comparator0. See Table 7.1 for complete timing and power consumption specifications.

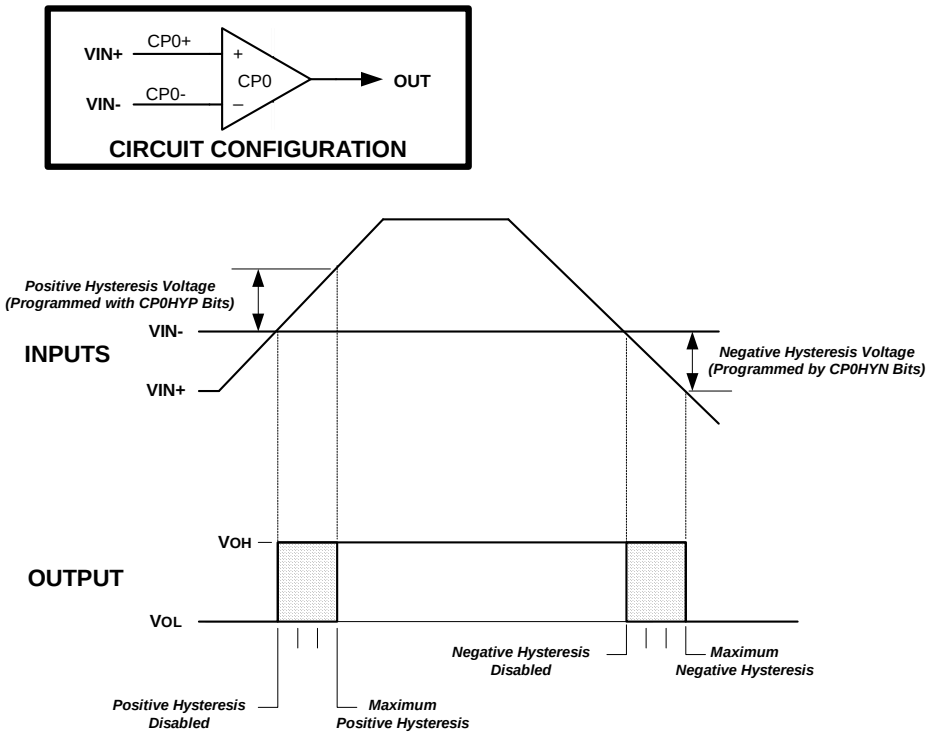


**Figure 7.2. Comparator Hysteresis Plot**

The hysteresis of Comparator0 is software-programmable via its Comparator0 Control register (CPT0CN). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator0 hysteresis is programmed using Bits3–0 in the Comparator0 Control Register CPT0CN (shown in SFR Definition 7.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 7.2, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

**Table 8.1. CIP-51 Instruction Set Summary (Continued)**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| MOVC A, @A+PC | Move code byte relative PC to A | 1 | 3 |
| MOVX A, @Ri | Move external data (8-bit address) to A | 1 | 3 |
| MOVX @Ri, A | Move A to external data (8-bit address) | 1 | 3 |
| MOVX A, @DPTR | Move external data (16-bit address) to A | 1 | 3 |
| MOVX @DPTR, A | Move A to external data (16-bit address) | 1 | 3 |
| PUSH direct | Push direct byte onto stack | 2 | 2 |
| POP direct | Pop direct byte from stack | 2 | 2 |
| XCH A, Rn | Exchange Register with A | 1 | 1 |
| XCH A, direct | Exchange direct byte with A | 2 | 2 |
| XCH A, @Ri | Exchange indirect RAM with A | 1 | 2 |
| XCHD A, @Ri | Exchange low nibble of indirect RAM with A | 1 | 2 |
| **Boolean Manipulation** | | | |
| CLR C | Clear Carry | 1 | 1 |
| CLR bit | Clear direct bit | 2 | 2 |
| SETB C | Set Carry | 1 | 1 |
| SETB bit | Set direct bit | 2 | 2 |
| CPL C | Complement Carry | 1 | 1 |
| CPL bit | Complement direct bit | 2 | 2 |
| ANL C, bit | AND direct bit to Carry | 2 | 2 |
| ANL C, /bit | AND complement of direct bit to Carry | 2 | 2 |
| ORL C, bit | OR direct bit to carry | 2 | 2 |
| ORL C, /bit | OR complement of direct bit to Carry | 2 | 2 |
| MOV C, bit | Move direct bit to Carry | 2 | 2 |
| MOV bit, C | Move Carry to direct bit | 2 | 2 |
| JC rel | Jump if Carry is set | 2 | 2/3 |
| JNC rel | Jump if Carry is not set | 2 | 2/3 |
| JB bit, rel | Jump if direct bit is set | 3 | 3/4 |
| JNB bit, rel | Jump if direct bit is not set | 3 | 3/4 |
| JBC bit, rel | Jump if direct bit is set and clear bit | 3 | 3/4 |
| **Program Branching** | | | |
| ACALL addr11 | Absolute subroutine call | 2 | 3 |
| LCALL addr16 | Long subroutine call | 3 | 4 |
| RET | Return from subroutine | 1 | 5 |
| RETI | Return from interrupt | 1 | 5 |
| AJMP addr11 | Absolute jump | 2 | 3 |
| LJMP addr16 | Long jump | 3 | 4 |
| SJMP rel | Short jump (relative address) | 2 | 3 |
| JMP @A+DPTR | Jump indirect relative to DPTR | 1 | 3 |
| JZ rel | Jump if A equals zero | 2 | 2/3 |

**Table 8.3. Special Function Registers* (Continued)**

| Register | Address | Description | Page No. |
|---|---|---|---|
| FLSCL | 0xB6 | Flash Scale | 93 |
| IE | 0xA8 | Interrupt Enable | 75 |
| IP | 0xB8 | Interrupt Priority | 76 |
| IT01CF | 0xE4 | INT0/INT1 Configuration Register | 79 |
| OSCICL | 0xB3 | Internal Oscillator Calibration | 98 |
| OSCICN | 0xB2 | Internal Oscillator Control | 98 |
| OSCXCN | 0xB1 | External Oscillator Control | 100 |
| P0 | 0x80 | Port 0 Latch | 109 |
| P0MDIN | 0xF1 | Port 0 Input Mode Configuration | 109 |
| P0MDOUT | 0xA4 | Port 0 Output Mode Configuration | 110 |
| PCA0CN | 0xD8 | PCA Control | 167 |
| PCA0MD | 0xD9 | PCA Mode | 168 |
| PCA0CPH0 | 0xFC | PCA Capture 0 High | 171 |
| PCA0CPH1 | 0xEA | PCA Capture 1 High | 171 |
| PCA0CPH2 | 0xEC | PCA Capture 2 High | 171 |
| PCA0CPL0 | 0xFB | PCA Capture 0 Low | 171 |
| PCA0CPL1 | 0xE9 | PCA Capture 1 Low | 171 |
| PCA0CPL2 | 0xEB | PCA Capture 2 Low | 171 |
| PCA0CPM0 | 0xDA | PCA Module 0 Mode Register | 169 |
| PCA0CPM1 | 0xDB | PCA Module 1 Mode Register | 169 |
| PCA0CPM2 | 0xDC | PCA Module 2 Mode Register | 169 |
| PCA0H | 0xFA | PCA Counter High | 170 |
| PCA0L | 0xF9 | PCA Counter Low | 170 |
| PCON | 0x87 | Power Control | 81 |
| PSCTL | 0x8F | Program Store R/W Control | 92 |
| PSW | 0xD0 | Program Status Word | 70 |
| REF0CN | 0xD1 | Voltage Reference Control | 49 |
| RSTSRC | 0xEF | Reset Source Configuration/Status | 87 |
| SBUF0 | 0x99 | UART 0 Data Buffer | 137 |
| SCON0 | 0x98 | UART 0 Control | 136 |
| SMB0CF | 0xC1 | SMBus Configuration | 118 |
| SMB0CN | 0xC0 | SMBus Control | 120 |
| SMB0DAT | 0xC2 | SMBus Data | 122 |
| SP | 0x81 | Stack Pointer | 69 |
| TMR2CN | 0xC8 | Timer/Counter 2 Control | 154 |
| TCON | 0x88 | Timer/Counter Control | 147 |
| TH0 | 0x8C | Timer/Counter 0 High | 150 |

***Note:** SFRs are listed in alphabetical order. All undefined SFR locations are reserved

## SFR Definition 8.4. PSW: Program Status Word

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CY | AC | F0 | RS1 | RS0 | OV | F1 | PARITY | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xD0 |

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: RS1-RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

| RS1 | RS0 | Register Bank | Address |
|---|---|---|---|
| 0 | 0 | 0 | 0x00–0x07 |
| 0 | 1 | 1 | 0x08–0x0F |
| 1 | 0 | 2 | 0x10–0x17 |
| 1 | 1 | 3 | 0x18–0x1F |

Bit2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
- An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
- A MUL instruction results in an overflow (result is greater than 255).
- A DIV instruction causes a divide-by-zero condition.

The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

## 8.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 12 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE-EIE1). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

**Note: Any instruction that clears the EA bit should be immediately followed by an instruction that has two or more opcode bytes. For example:**

```
// in 'C':
EA = 0;    // clear EA bit
EA = 0;    // ... followed by another 2-byte opcode

; in assembly:
CLR  EA    ; clear EA bit
CLR  EA    ; ... followed by another 2-byte opcode
```

If an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction which clears the EA bit), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. However, a read of the EA bit will return a '0' inside the interrupt service routine. When the "CLR EA" opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will reenter the ISR after the completion of the next instruction.

### 8.3.1. MCU Interrupt Sources and Vectors

The MCUs support 12 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 8.4 on page 74. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

**Table 11.1. Internal Oscillator Electrical Characteristics**

–40 to +85 °C unless otherwise specified

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Calibrated Internal Oscillator Frequency | C8051F300/1 devices<br>–40 to +85 °C | 24 | 24.5 | 25 | MHz |
| | C8051F300/1 devices<br>0 to +70 °C | 24.3 | 24.7 | 25 | MHz |
| Uncalibrated Internal Oscillator Frequency | C8051F302/3/4/5 devices | 16 | 20 | 24 | MHz |
| Internal Oscillator Supply Current (from $V_{DD}$) | OSCICN.2 = 1 | | 450 | | µA |

## 11.2. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 11.1. A 10 MΩ resistor also must be wired across the XTAL2 and XTAL1 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 11.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 11.3).

**Important Note on External Oscillator Usage:** Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are occupied as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is occupied as XTAL2. The Port I/O Crossbar should be configured to skip the occupied Port pins; see **Section “12.1. Priority Crossbar Decoder” on page 104** for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See **Section “12.2. Port I/O Initialization” on page 106** for details on Port input mode selection.

## 11.3. System Clock Selection

The CLKSL bit in register OSCICN selects which oscillator is used as the system clock. CLKSL must be set to ‘1’ for the system clock to run from the external oscillator; however the external oscillator may still clock 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 and external oscillator, so long as the selected oscillator is enabled and has settled. The internal oscillator requires little start-up time and may be enabled and selected as the system clock in the same write to OSCICN. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use as the system clock. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to ‘1’ by hardware when the external oscillator is settled. To avoid reading a false XTLVLD, in crystal mode software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD. RC and C modes typically require no start-up time.

## SFR Definition 11.3. OSCXCN: External Oscillator Control

| R | R/W | R/W | R/W | R | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| XTLVLD | XOSCMD2 | XOSCMD1 | XOSCMD0 | — | XFCN2 | XFCN1 | XFCN0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | | 0xB1 |

Bit7: XTLVLD: Crystal Oscillator Valid Flag.
(Read only when XOSCMD = 11x.)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits6–4: XOSCMD2-0: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode.
011: External CMOS Clock Mode with divide by 2 stage.
100: RC Oscillator Mode with divide by 2 stage.
101: Capacitor Oscillator Mode with divide by 2 stage.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit3: RESERVED. Read = 0, Write = don't care.

Bits2–0: XFCN2-0: External Oscillator Frequency Control Bits.
000-111: See table below:

| XFCN | Crystal (XOSCMD = 11x) | RC (XOSCMD = 10x) | C (XOSCMD = 10x) |
|---|---|---|---|
| 000 | f ≤ 32 kHz | f ≤ 25 kHz | K Factor = 0.87 |
| 001 | 32 kHz < f ≤ 84 kHz | 25 kHz < f ≤ 50 kHz | K Factor = 2.6 |
| 010 | 84 kHz < f ≤ 225 kHz | 50 kHz < f ≤ 100 kHz | K Factor = 7.7 |
| 011 | 225 kHz < f ≤ 590 kHz | 100 kHz < f ≤ 200 kHz | K Factor = 22 |
| 100 | 590 kHz < f ≤ 1.5 MHz | 200 kHz < f ≤ 400 kHz | K Factor = 65 |
| 101 | 1.5 MHz < f ≤ 4 MHz | 400 kHz < f ≤ 800 kHz | K Factor = 180 |
| 110 | 4 MHz < f ≤ 10 MHz | 800 kHz < f ≤ 1.6 MHz | K Factor = 664 |
| 111 | 10 MHz < f ≤ 30 MHz | 1.6 MHz < f ≤ 3.2 MHz | K Factor = 1590 |

**CRYSTAL MODE** (Circuit from Figure 11.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal frequency.

**RC MODE** (Circuit from Figure 11.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
**$f = 1.23(10^3) / (R \times C)$**, where
f = frequency of oscillation in MHz
C = capacitor value in pF
R = Pull-up resistor value in kΩ

**C MODE** (Circuit from Figure 11.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
**$f = KF / (C \times V_{DD})$**, where
f = frequency of oscillation in MHz
C = capacitor value the XTAL2 pin in pF
$V_{DD}$ = Power Supply on MCU in volts

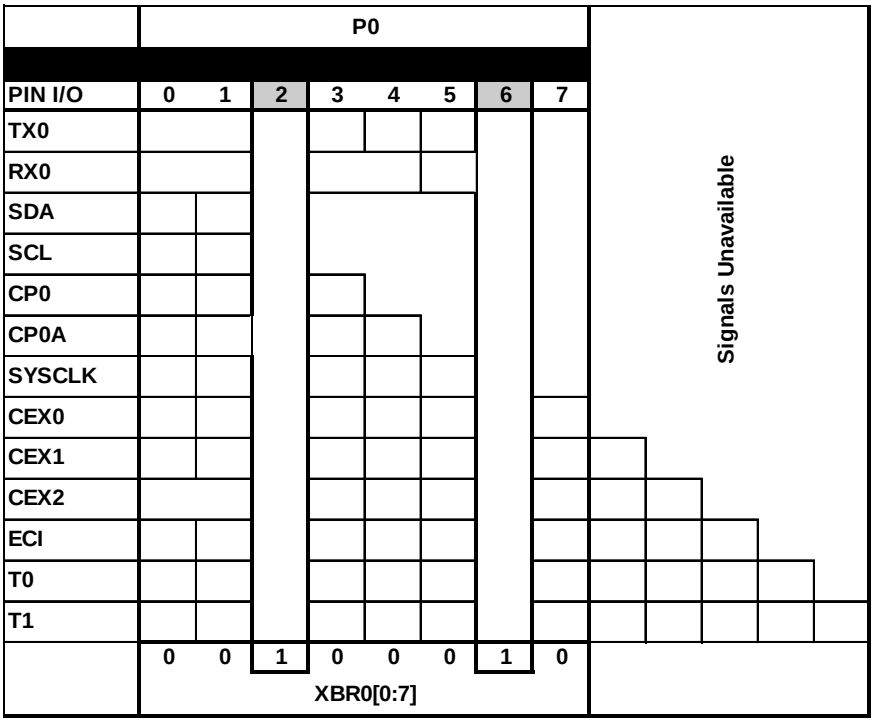


**Figure 12.4. Crossbar Priority Decoder with XBR0 = 0x44**

Registers XBR1 and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL). Either or both of the UART signals may be selected by the Crossbar. UART0 pin assignments are fixed for bootloading purposes: when UART TX0 is selected, it is always assigned to P0.4; when UART RX0 is selected, it is always assigned to P0.5. Standard Port I/Os appear contiguously after the prioritized functions have been assigned. For example, if assigned functions that take the first 3 Port I/O (P0.[2:0]), 5 Port I/O are left for analog or GPIO use.

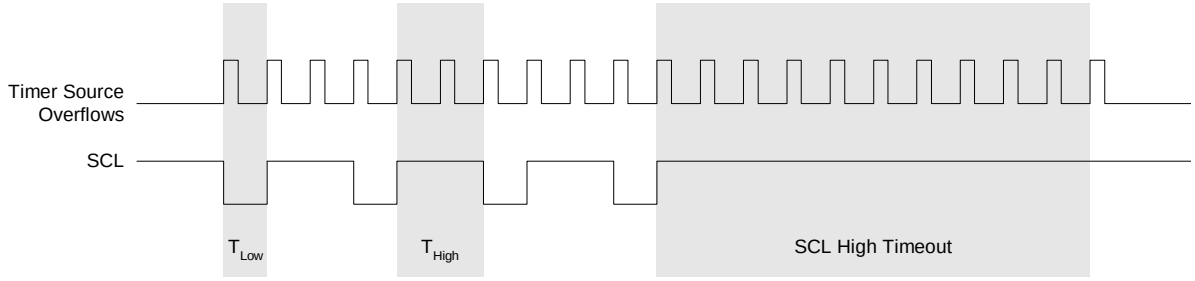


**Figure 13.4. Typical SMBus SCL Generation**

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 13.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

**Table 13.2. Minimum SDA Setup and Hold Times**

| EXTHOLD | Minimum SDA Setup Time | Minimum SDA Hold Time |
|---|---|---|
| 0 | $T_{low}$ – 4 system clocks<br>OR<br>1 system clock + s/w delay* | 3 system clocks |
| 1 | 11 system clocks | 12 system clocks |

***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. The s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.

With the SMBTOE bit set, Timer 2 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see **Section “13.3.3. SCL Low Timeout” on page 114**). The SMBus interface will force Timer 2 to reload while SCL is high, and allow Timer 2 to count when SCL is low. The Timer 2 interrupt service routine should be used to reset SMBus communication by disabling and reenabling the SMBus. Timer 2 configuration is described in **Section “15.2. Timer 2” on page 151**.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 13.4). When a Free Timeout is detected, the interface will respond as if a STOP was detected (an interrupt will be generated, and STO will be set).

**Table 13.4. SMBus Status Decoding (Continued)**

| Mode | Values Read | | | | Current SMbus State | Typical Response Options | Values Written | | |
|---|---|---|---|---|---|---|---|---|---|
| | Status Vector | ACKRQ | ARBLOST | ACK | | | STA | STO | ACK |
| SLAVE RECEIVER | 0010 | 1 | 0 | X | A slave address was received; ACK requested. | Acknowledge received address (received slave address match, R/W bit = READ). | 0 | 0 | 1 |
| | | | | | | Do not acknowledge received address. | 0 | 0 | 0 |
| | | | | | | Acknowledge received address, and switch to transmitter mode (received slave address match, R/W bit = WRITE); see **Section 13.5.4** for procedure. | 0 | 0 | 1 |
| | | 1 | 1 | X | Lost arbitration as master; slave address received; ACK requested. | Acknowledge received address (received slave address match, R/W bit = READ). | 0 | 0 | 1 |
| | | | | | | Do not acknowledge received address. | 0 | 0 | 0 |
| | | | | | | Acknowledge received address, and switch to transmitter mode (received slave address match, R/W bit = WRITE); see **Section 13.5.4** for procedure. | 0 | 0 | 1 |
| | | | | | | Reschedule failed transfer; do not acknowledge received address | 1 | 0 | 0 |
| | 0010 | 0 | 1 | X | Lost arbitration while attempting a repeated START. | Abort failed transfer. | 0 | 0 | X |
| | | | | | | Reschedule failed transfer. | 1 | 0 | X |
| | 0001 | 1 | 1 | X | Lost arbitration while attempting a STOP. | No action required (transfer complete/aborted). | 0 | 0 | 0 |
| | | 0 | 0 | X | A STOP was detected while addressed as a Slave Transmitter or Slave Receiver. | Clear STO. | 0 | 0 | X |
| | | 0 | 1 | X | Lost arbitration due to a detected STOP. | Abort transfer. | 0 | 0 | X |
| | | | | | | Reschedule failed transfer. | 1 | 0 | X |
| | 0000 | 1 | 0 | X | A slave byte was received; ACK requested. | Acknowledge received byte; Read SMB0DAT. | 0 | 0 | 1 |
| | | | | | | Do not acknowledge received byte. | 0 | 0 | 0 |
| | | 1 | 1 | X | Lost arbitration while transmitting a data byte as master. | Abort failed transfer. | 0 | 0 | 0 |
| | | | | | | Reschedule failed transfer. | 1 | 0 | 0 |

# 14. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in **Section "14.1. Enhanced Baud Rate Generation" on page 132**). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. Reading SBUF0 accesses the buffered Receive register; writing SBUF0 accesses the Transmit register.

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

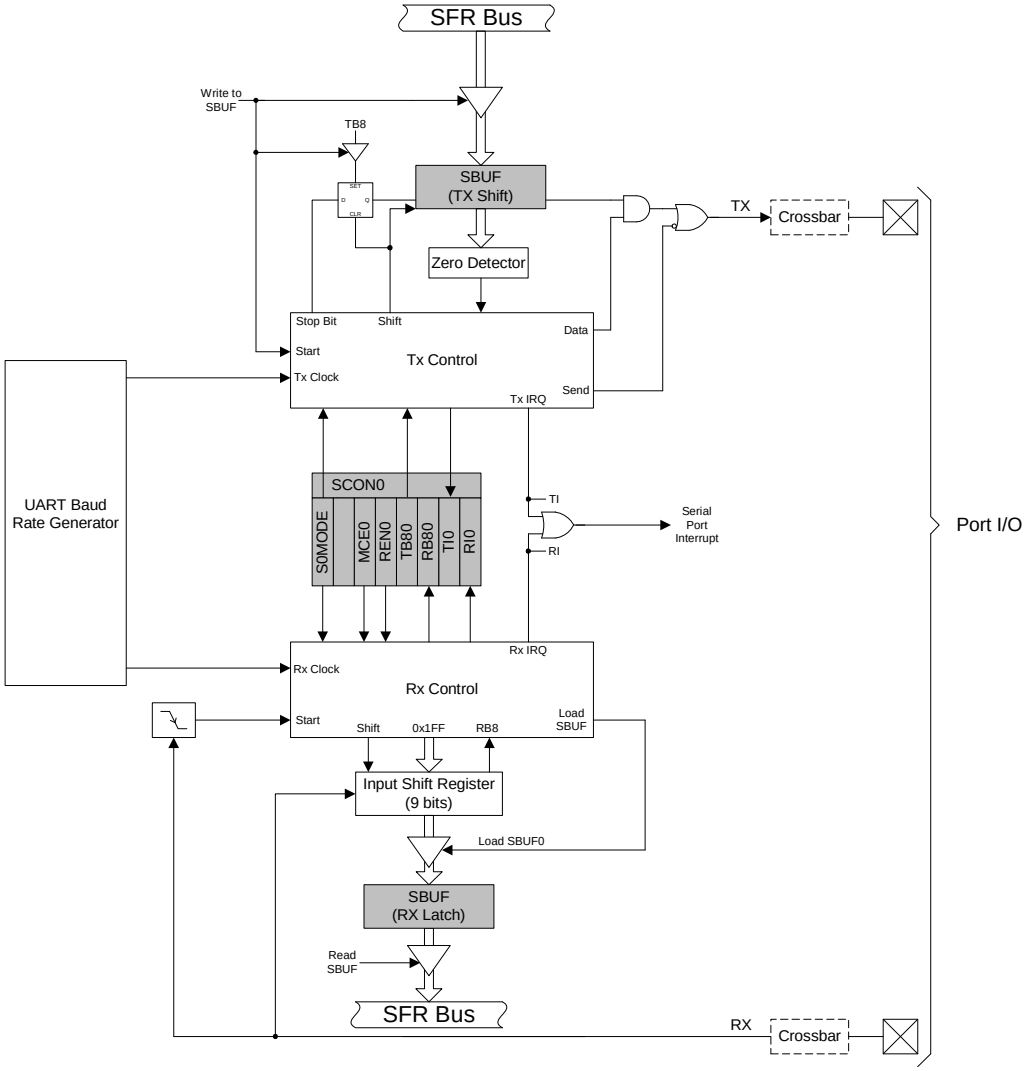


**Figure 14.1. UART0 Block Diagram**

## SFR Definition 14.2. SBUF0: Serial (UART0) Port Data Buffer

| 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: |
| | | | | | | | | 0x99 |

Bits7–0: SBUF0[7:0]: Serial Data Buffer Bits 7–0 (MSB-LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 is what initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

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

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

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

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal /INT0 is active as defined by bit IN0PL in register IT01CF (see **Section “8.3.2. External Interrupts” on page 73** for details on the external input signals /INT0 and /INT1).

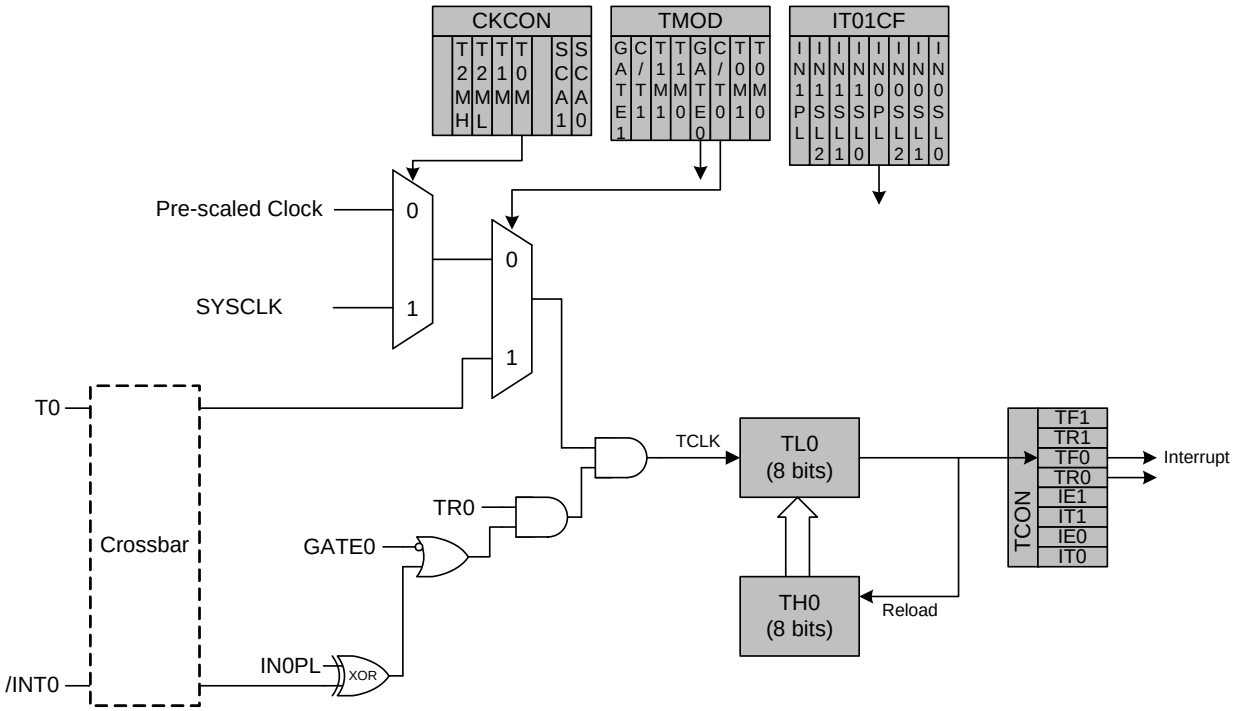


**Figure 15.2. T0 Mode 2 Block Diagram**

### 15.1.4. 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.

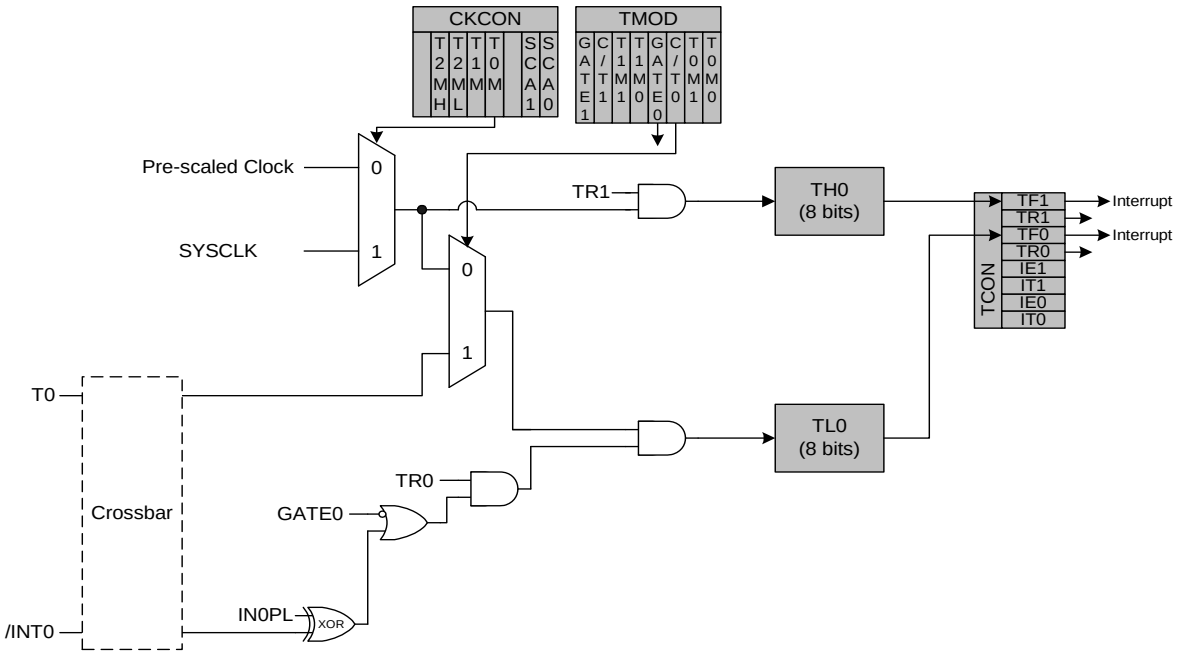


**Figure 15.3. T0 Mode 3 Block Diagram**

### 16.2.3. 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) Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

**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'.

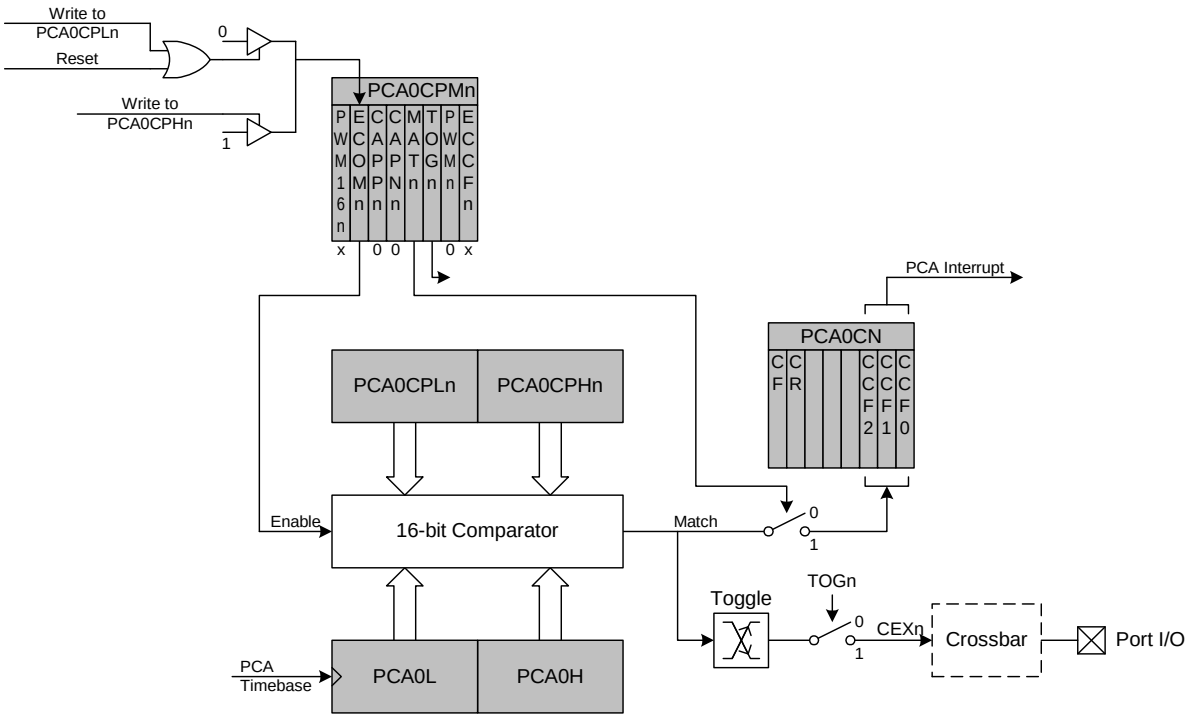


**Figure 16.6. PCA High Speed Output Mode Diagram**

## NOTES: