**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 | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 48 MIPS |
| Connectivity | EBI/EMI, I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT |
| Number of I/O | 40 |
| Program Memory Size | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.25V |
| Data Converters | A/D 32x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-TQFP |
| Supplier Device Package | 48-TQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f388-b-gq |

**Table 3.1. Pin Definitions for the C8051F388/9/A/B (Continued)**

| Name | Pin Numbers | | Type | Description |
|---|---|---|---|---|
| | 48-pin | 32-pin | | |
| P3.2 | 28 | — | D I/O or A In | Port 3.2. |
| P3.3 | 27 | — | D I/O or A In | Port 3.3. |
| P3.4 | 26 | — | D I/O or A In | Port 3.4. |
| P3.5 | 25 | — | D I/O or A In | Port 3.5. |
| P3.6 | 24 | — | D I/O or A In | Port 3.6. |
| P3.7 | 23 | — | D I/O or A In | Port 3.7. |
| P4.0 | 22 | — | D I/O or A In | Port 4.0. See Section 20 for a complete description of Port 4. |
| P4.1 | 21 | — | D I/O or A In | Port 4.1. |
| P4.2 | 20 | — | D I/O or A In | Port 4.2. |
| P4.3 | 19 | — | D I/O or A In | Port 4.3. |
| P4.4 | 18 | — | D I/O or A In | Port 4.4. |
| P4.5 | 17 | — | D I/O or A In | Port 4.5. |
| P4.6 | 16 | — | D I/O or A In | Port 4.6. |
| P4.7 | 15 | — | D I/O or A In | Port 4.7. |

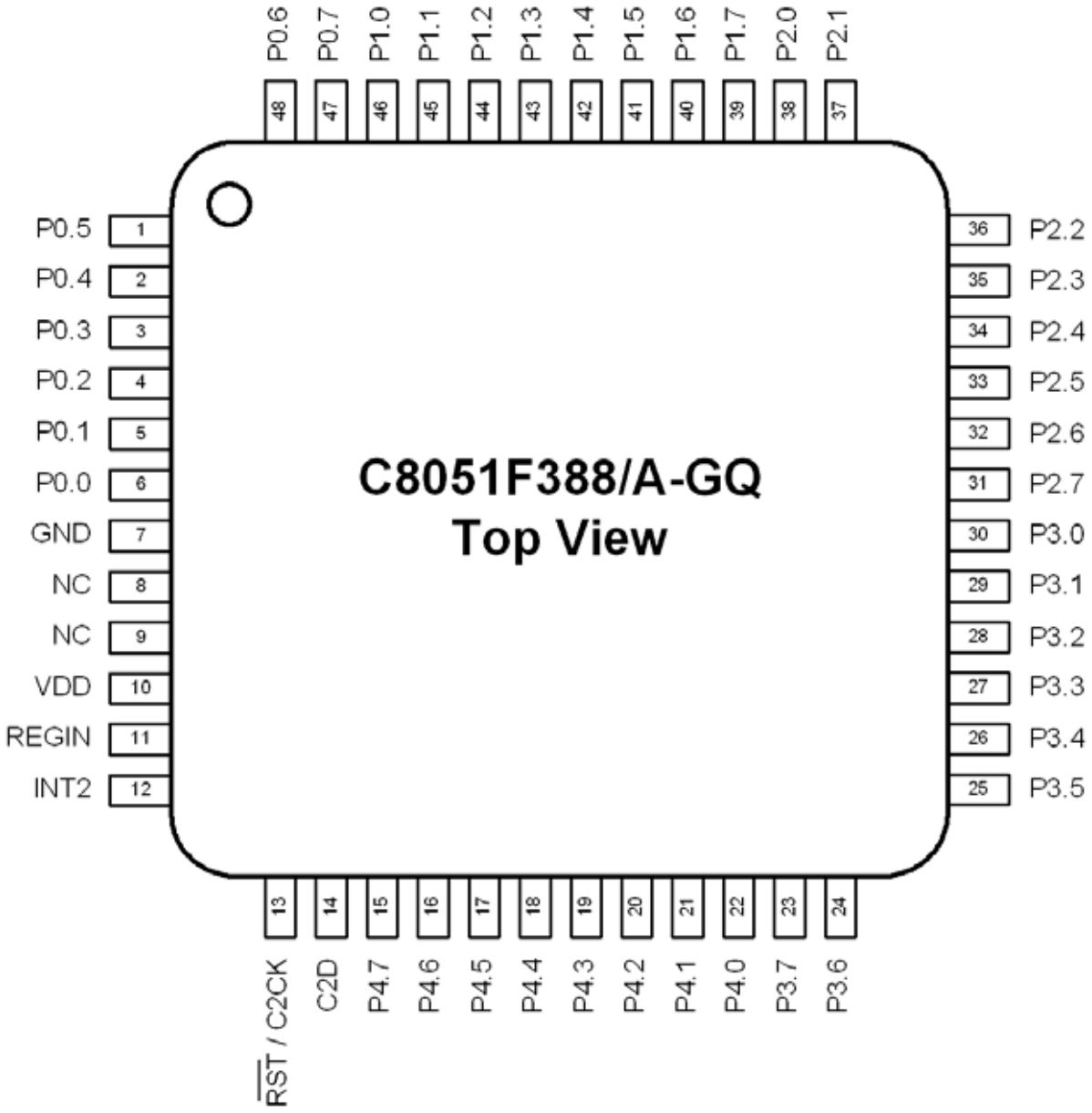


**Figure 3.1. TQFP-48 (C8051F388/A) Pinout Diagram (Top View)**

### Table 5.7. Internal High-Frequency Oscillator Electrical Characteristics

$V_{DD}$ = 2.7 to 3.6 V; $T_A$ = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

| Parameter | Test Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Oscillator Frequency | IFCN = 11b | 47.3 | 48 | 48.7 | MHz |
| Oscillator Supply Current (from $V_{DD}$) | 25 °C, $V_{DD}$ = 3.0 V, OSCICN.7 = 1, OCSICN.5 = 0 | — | 900 | — | µA |
| Power Supply Sensitivity | Constant Temperature | — | 110 | — | ppm/V |
| Temperature Sensitivity | Constant Supply | — | 25 | — | ppm/°C |

### Table 5.8. Internal Low-Frequency Oscillator Electrical Characteristics

$V_{DD}$ = 2.7 to 3.6 V; $T_A$ = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

| Parameter | Test Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Oscillator Frequency | OSCLD = 11b | 75 | 80 | 85 | kHz |
| Oscillator Supply Current (from $V_{DD}$) | 25 °C, $V_{DD}$ = 3.0 V, OSCLCN.7 = 1 | — | 4 | — | µA |
| Power Supply Sensitivity | Constant Temperature | — | 0.05 | — | %/V |
| Temperature Sensitivity | Constant Supply | — | 65 | — | ppm/°C |

### Table 5.9. External Oscillator Electrical Characteristics

$V_{DD}$ = 2.7 to 3.6 V; $T_A$ = –40 to +85 °C unless otherwise specified.

| Parameter | Test Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| External Crystal Frequency | | 0.02 | — | 30 | MHz |
| External CMOS Oscillator Frequency | | 0 | — | 48 | MHz |

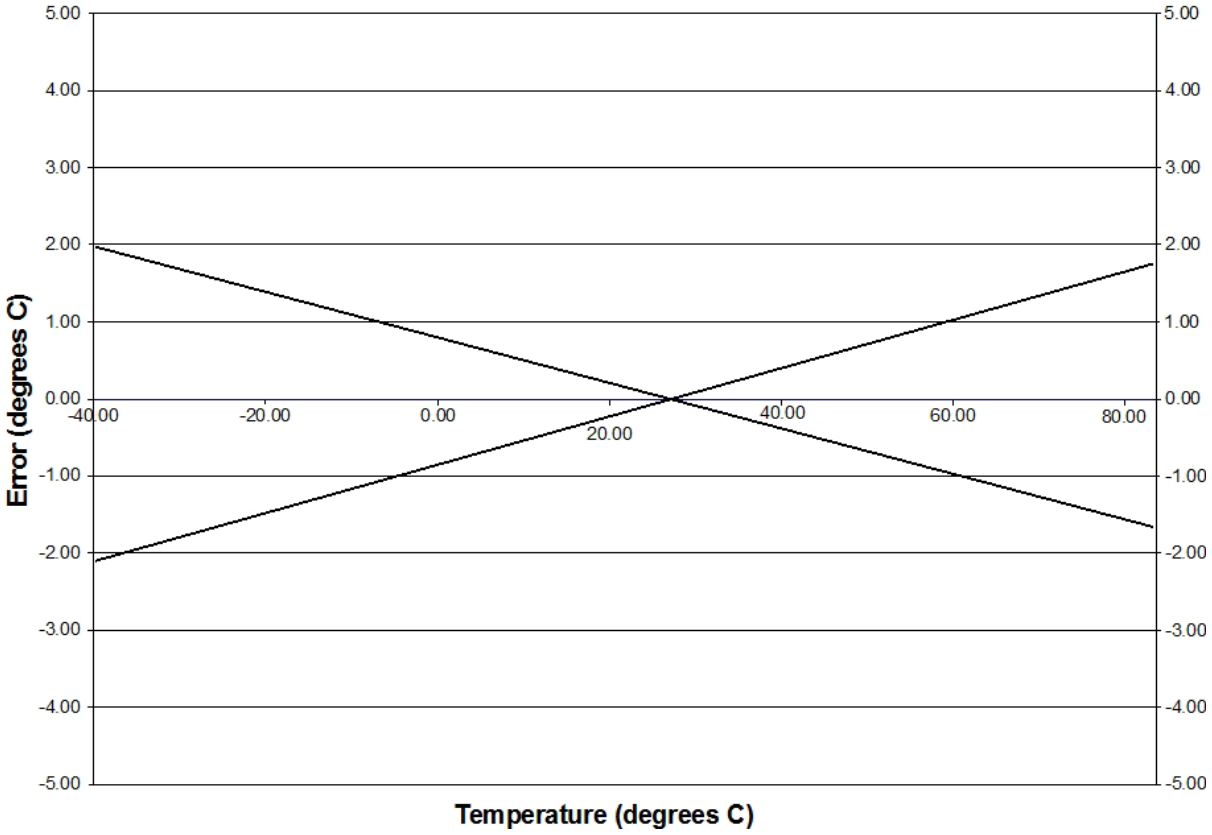


**Figure 6.3. Temperature Sensor Error with 1-Point Calibration**

## 11.2. CIP-51 Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should always be written to the value indicated in the SFR description. Future product versions may use these bits to implement new features in which case the reset value of the bit will be the indicated value, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

### SFR Definition 11.1. DPL: Data Pointer Low Byte

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

SFR Address = 0x82; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | DPL[7:0] | **Data Pointer Low.**<br>The DPL register is the low byte of the 16-bit DPTR. |

### SFR Definition 11.2. DPH: Data Pointer High Byte

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

SFR Address = 0x83; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | DPH[7:0] | **Data Pointer High.**<br>The DPH register is the high byte of the 16-bit DPTR. |

## SFR Definition 11.6. PSW: Program Status Word

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CY | AC | F0 | RS[1:0] | | OV | F1 | PARITY |
| Type | R/W | R/W | R/W | R/W | | R/W | R/W | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xD0; SFR Page = All Pages; Bit-Addressable

| Bit | Name | Function |
|---|---|---|
| 7 | CY | **Carry Flag.**<br>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. |
| 6 | AC | **Auxiliary Carry Flag.**<br>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. |
| 5 | F0 | **User Flag 0.**<br>This is a bit-addressable, general purpose flag for use under software control. |
| 4:3 | RS[1:0] | **Register Bank Select.**<br>These bits select which register bank is used during register accesses.<br>00: Bank 0, Addresses 0x00-0x07<br>01: Bank 1, Addresses 0x08-0x0F<br>10: Bank 2, Addresses 0x10-0x17<br>11: Bank 3, Addresses 0x18-0x1F |
| 2 | OV | **Overflow Flag.**<br>This bit is set to 1 under the following circumstances:<br>• An ADD, ADDC, or SUBB instruction causes a sign-change overflow.<br>• A MUL instruction results in an overflow (result is greater than 255).<br>• A DIV instruction causes a divide-by-zero condition.<br>The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases. |
| 1 | F1 | **User Flag 1.**<br>This is a bit-addressable, general purpose flag for use under software control. |
| 0 | PARITY | **Parity Flag.**<br>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. |

## 14.4. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMI0CF.4) bit.

### 14.4.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 14.1.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the 'Q' outputs reflect the states of the 'D' inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time $\overline{RD}$ or $\overline{WR}$ is asserted.

See Section "14.6.2. Multiplexed Mode" on page 101 for more information.

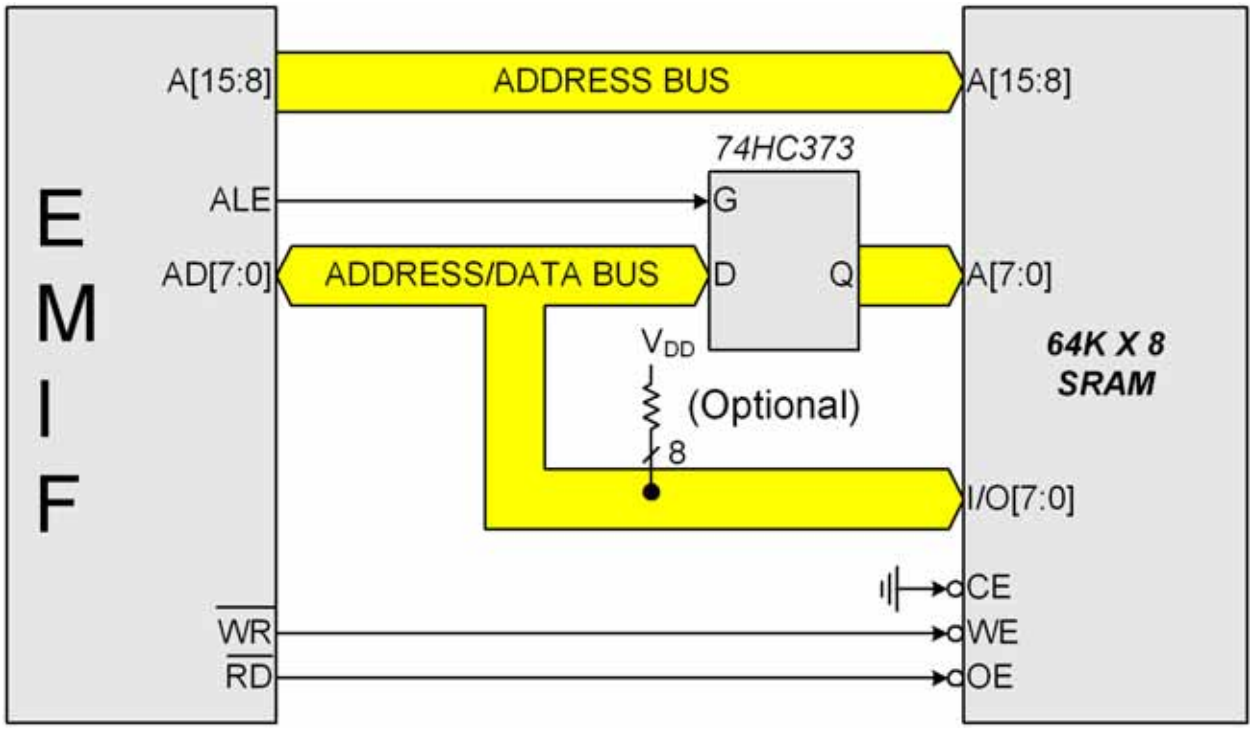


**Figure 14.1. Multiplexed Configuration Example**

**Table 14.1. AC Parameters for External Memory Interface**

| Parameter | Description | Min* | Max* | Units |
|---|---|---|---|---|
| $T_{ACS}$ | Address/Control Setup Time | 0 | 3 x $T_{SYSCLK}$ | ns |
| $T_{ACW}$ | Address/Control Pulse Width | 1 x $T_{SYSCLK}$ | 16 x $T_{SYSCLK}$ | ns |
| $T_{ACH}$ | Address/Control Hold Time | 0 | 3 x $T_{SYSCLK}$ | ns |
| $T_{ALEH}$ | Address Latch Enable High Time | 1 x $T_{SYSCLK}$ | 4 x $T_{SYSCLK}$ | ns |
| $T_{ALEL}$ | Address Latch Enable Low Time | 1 x $T_{SYSCLK}$ | 4 x $T_{SYSCLK}$ | ns |
| $T_{WDS}$ | Write Data Setup Time | 1 x $T_{SYSCLK}$ | 19 x $T_{SYSCLK}$ | ns |
| $T_{WDH}$ | Write Data Hold Time | 0 | 3 x $T_{SYSCLK}$ | ns |
| $T_{RDS}$ | Read Data Setup Time | 20 | | ns |
| $T_{RDH}$ | Read Data Hold Time | 0 | | ns |
| **Note:** $T_{SYSCLK}$ is equal to one period of the device system clock (SYSCLK). | | | | |

**Table 15.2. Special Function Registers (Continued)**

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

| Register | Address | Page | Description | Page |
|---|---|---|---|---|
| **SMB1CN** | 0xC0 | F | SMBus1 Control | 177 |
| **SMB1DAT** | 0xC2 | F | SMBus1 Data | 183 |
| **SMBTC** | 0xB9 | F | SMBus0/1 Timing Control | 174 |
| **SMOD1** | 0xE5 | All Pages | UART1 Mode | 207 |
| **SP** | 0x81 | All Pages | Stack Pointer | 82 |
| **SPI0CFG** | 0xA1 | All Pages | SPI Configuration | 218 |
| **SPI0CKR** | 0xA2 | All Pages | SPI Clock Rate Control | 220 |
| **SPI0CN** | 0xF8 | All Pages | SPI Control | 219 |
| **SPI0DAT** | 0xA3 | All Pages | SPI Data | 220 |
| **TCON** | 0x88 | All Pages | Timer/Counter Control | 231 |
| **TH0** | 0x8C | All Pages | Timer/Counter 0 High | 234 |
| **TH1** | 0x8D | All Pages | Timer/Counter 1 High | 234 |
| **TL0** | 0x8A | All Pages | Timer/Counter 0 Low | 233 |
| **TL1** | 0x8B | All Pages | Timer/Counter 1 Low | 233 |
| **TMOD** | 0x89 | All Pages | Timer/Counter Mode | 232 |
| **TMR2CN** | 0xC8 | 0 | Timer/Counter 2 Control | 239 |
| **TMR2H** | 0xCD | 0 | Timer/Counter 2 High | 241 |
| **TMR2L** | 0xCC | 0 | Timer/Counter 2 Low | 240 |
| **TMR2RLH** | 0xCB | 0 | Timer/Counter 2 Reload High | 240 |
| **TMR2RLL** | 0xCA | 0 | Timer/Counter 2 Reload Low | 240 |
| **TMR3CN** | 0x91 | 0 | Timer/Counter 3 Control | 246 |
| **TMR3H** | 0x95 | 0 | Timer/Counter 3 High | 248 |
| **TMR3L** | 0x94 | 0 | Timer/Counter 3 Low | 247 |
| **TMR3RLH** | 0x93 | 0 | Timer/Counter 3 Reload High | 247 |
| **TMR3RLL** | 0x92 | 0 | Timer/Counter 3 Reload Low | 247 |
| **TMR4CN** | 0x91 | F | Timer/Counter 4 Control | 251 |
| **TMR4H** | 0x95 | F | Timer/Counter 4 High | 253 |
| **TMR4L** | 0x94 | F | Timer/Counter 4 Low | 252 |
| **TMR4RLH** | 0x93 | F | Timer/Counter 4 Reload High | 252 |
| **TMR4RLL** | 0x92 | F | Timer/Counter 4 Reload Low | 252 |
| **TMR5CN** | 0xC8 | F | Timer/Counter 5 Control | 256 |
| **TMR5H** | 0xCD | F | Timer/Counter 5 High | 258 |
| **TMR5L** | 0xCC | F | Timer/Counter 5 Low | 257 |
| **TMR5RLH** | 0xCB | F | Timer/Counter 5 Reload High | 257 |

## 17.2. Power-Fail Reset / $V_{DD}$ Monitor

When a powerdown transition or power irregularity causes $V_{DD}$ to drop below $V_{RST}$, the power supply monitor will drive the $\overline{RST}$ pin low and hold the CIP-51 in a reset state (see Figure 17.2). When $V_{DD}$ returns to a level above $V_{RST}$, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if $V_{DD}$ dropped below the level required for data retention. If the PORSF flag reads 1, the data may no longer be valid. The $V_{DD}$ monitor is enabled after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the $V_{DD}$ monitor is disabled by code and a software reset is performed, the $V_{DD}$ monitor will still be disabled after the reset.

I**mportant Note:** If the $V_{DD}$ monitor is being turned on from a disabled state, it should be enabled before it is selected as a reset source. Selecting the $V_{DD}$ monitor as a reset source before it is enabled and stabilized may cause a system reset. In some applications, this reset may be undesirable. If this is not desirable in the application, a delay should be introduced between enabling the monitor and selecting it as a reset source. The procedure for enabling the $V_{DD}$ monitor and configuring it as a reset source from a disabled state is shown below:

1. Enable the $V_{DD}$ monitor (VDMEN bit in VDM0CN = 1).
2. If necessary, wait for the $V_{DD}$ monitor to stabilize (see Table 5.4 for the $V_{DD}$ Monitor turn-on time).
3. Select the $V_{DD}$ monitor as a reset source (PORSF bit in RSTSRC = 1).

See Figure 17.2 for $V_{DD}$ monitor timing; note that the power-on-reset delay is not incurred after a $V_{DD}$ monitor reset. See Table 5.4 for complete electrical characteristics of the $V_{DD}$ monitor.

## SFR Definition 19.3. OSCICN: Internal H-F Oscillator Control

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

SFR Address = 0xB2; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7 | IOSCEN | **Internal H-F Oscillator Enable Bit.**<br>0: Internal H-F Oscillator Disabled.<br>1: Internal H-F Oscillator Enabled. |
| 6 | IFRDY | **Internal H-F Oscillator Frequency Ready Flag.**<br>0: Internal H-F Oscillator is not running at programmed frequency.<br>1: Internal H-F Oscillator is running at programmed frequency. |
| 5 | SUSPEND | **Internal Oscillator Suspend Enable Bit.**<br>Setting this bit to logic 1 places the internal oscillator in SUSPEND mode. The internal oscillator resumes operation when one of the SUSPEND mode awakening events occurs. |
| 4:2 | Unused | Read = 000b; Write = don't care |
| 1:0 | IFCN[1:0] | **Internal H-F Oscillator Frequency Divider Control Bits.**<br>The Internal H-F Oscillator is divided by the IFCN bit setting after a divide-by-4 stage.<br>00: SYSCLK can be derived from Internal H-F Oscillator divided by 8 (1.5 MHz).<br>01: SYSCLK can be derived from Internal H-F Oscillator divided by 4 (3 MHz).<br>10: SYSCLK can be derived from Internal H-F Oscillator divided by 2 (6 MHz).<br>11: SYSCLK can be derived from Internal H-F Oscillator divided by 1 (12 MHz). |

## 19.3. Clock Multiplier

The C8051F388/9/A/B device includes a 48 MHz high-frequency oscillator instead of a 12 MHz oscillator and a 4x Clock Multiplier. For compatibility with C8051F34x and C8051F32x devices however, the CLKMUL register (SFR Definition 19.4) behaves as if the Clock Multiplier is present and working.

### SFR Definition 19.4. CLKMUL: Clock Multiplier Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MULEN | MULINIT | MULRDY | | | | MULSEL[1:0] | |
| Type | R | R | R | R | R | R | R | |
| Reset | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xB9; SFR Page = 0

| Bit | Name | Description |
|---|---|---|
| 7 | MULEN | **Clock Multiplier Enable Bit.**<br>This bit always reads 1. |
| 6 | MULINIT | **Clock Multiplier Initialize Bit.**<br>This bit always reads 1. |
| 5 | MULRDY | **Clock Multiplier Ready Bit.**<br>This bit always reads 1. |
| 4:2 | Unused | Read = 000b; Write = don't care |
| 1:0 | MULSEL[1:0] | **Clock Multiplier Input Select Bits.**<br>These bits always read 00. |

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 21.3 illustrates a typical SMBus transaction.

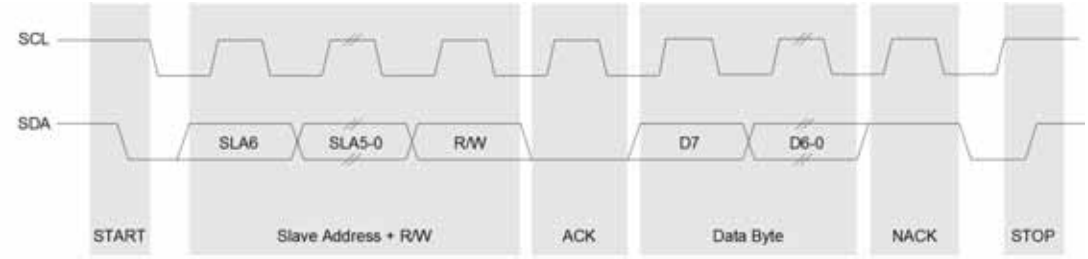


**Figure 21.3. SMBus Transaction**

### 21.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

### 21.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “21.3.5. SCL High (SMBus Free) Timeout” on page 169). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

### 21.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

### 21.3.4. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

For the SMBus0 interface, Timer 3 is used to implement SCL low timeouts. Timer 4 is used on the SMBus1 interface for SCL low timeouts. The SCL low timeout feature is enabled by setting the SMBnTOE bit in SMBnCF. The associated timer is forced to reload when SCL is high, and allowed to count when SCL is

## 22. 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 “22.1. Enhanced Baud Rate Generation” on page 194). 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. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from 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).

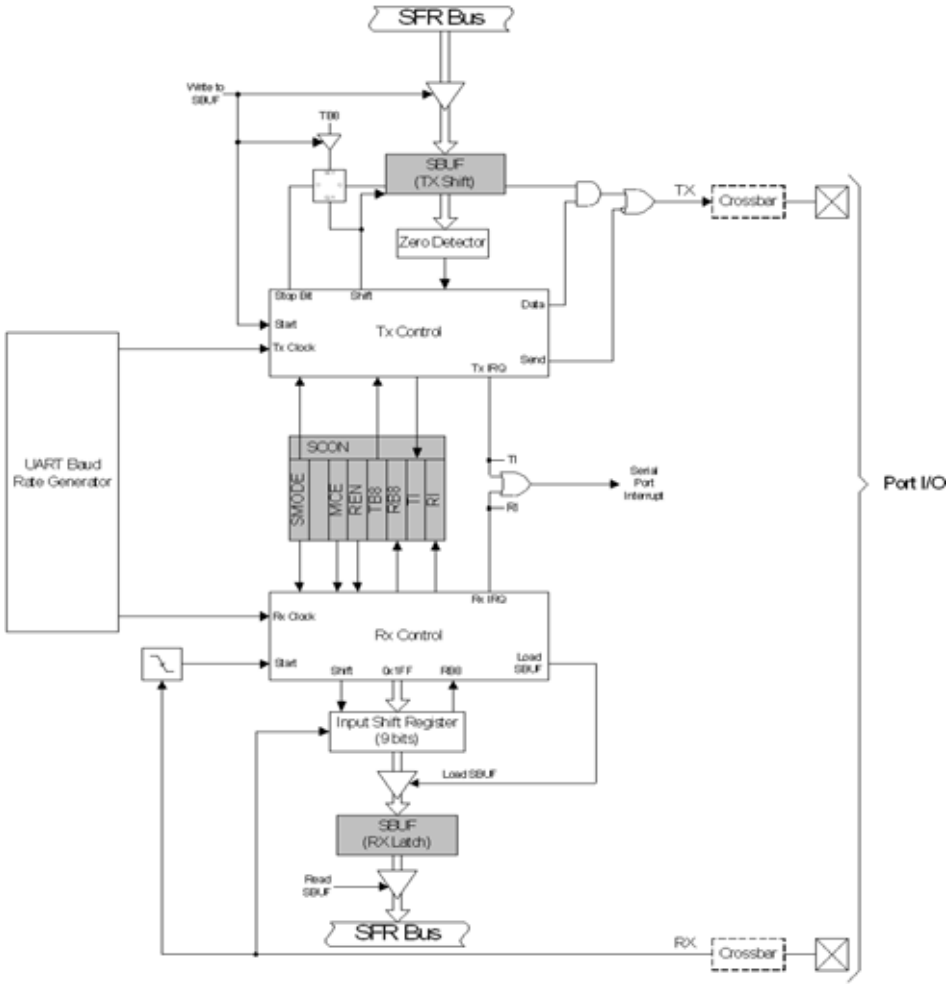


**Figure 22.1. UART0 Block Diagram**

1. Clear RI1 to 0
2. Read SBUF1
3. Check RI1, and repeat at Step 1 if RI1 is set to 1.

If the extra bit function is enabled (XBE1 = 1) and the parity function is disabled (PE1 = 0), the extra bit for the oldest byte in the FIFO can be read from the RBX1 bit (SCON1.2). If the extra bit function is not enabled, the value of the stop bit for the oldest FIFO byte will be presented in RBX1. When the parity function is enabled (PE1 = 1), hardware will check the received parity bit against the selected parity type (selected with S1PT[1:0]) when receiving data. If a byte with parity error is received, the PERR1 flag will be set to 1. This flag must be cleared by software. Note: when parity is enabled, the extra bit function is not available.

### 23.3.3. Multiprocessor Communications

UART1 supports multiprocessor communication between a master processor and one or more slave processors by special use of the extra data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its extra bit is logic 1; in a data byte, the extra bit is always set to logic 0.

Setting the MCE1 bit (SMOD1.7) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the extra bit is logic 1 (RBX1 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned address. If the addresses match, the slave will clear its MCE1 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE1 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE1 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

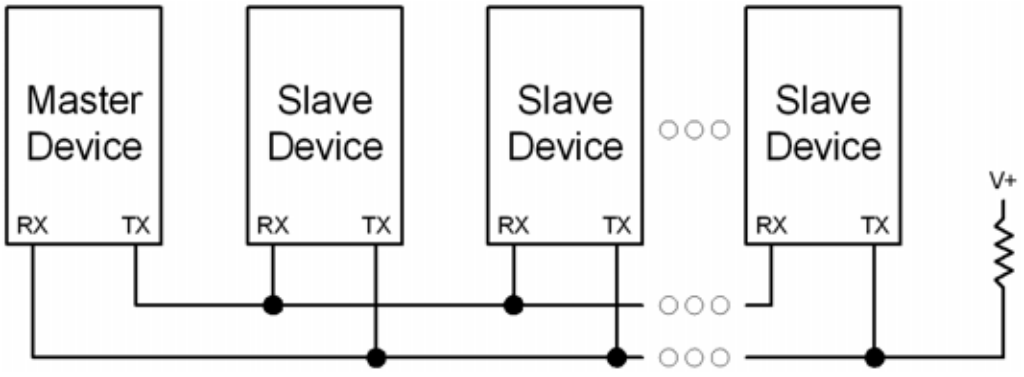


**Figure 23.6. UART Multi-Processor Mode Interconnect Diagram**

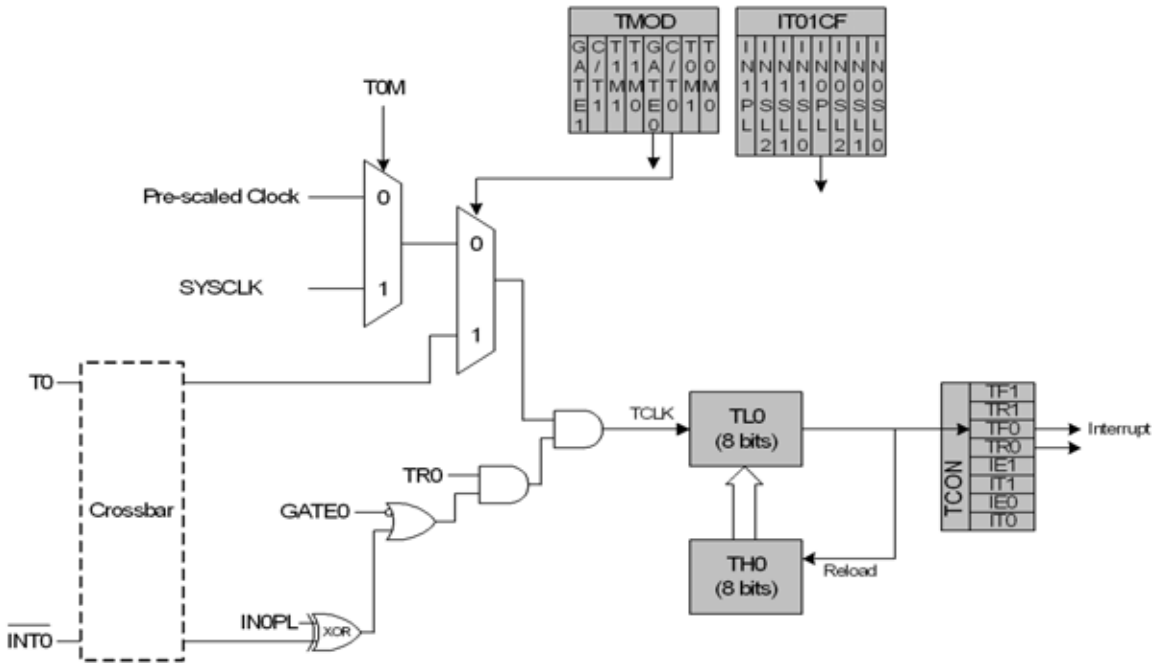


**Figure 25.2. T0 Mode 2 Block Diagram**

### 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 or overflow conditions for other peripherals. 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.

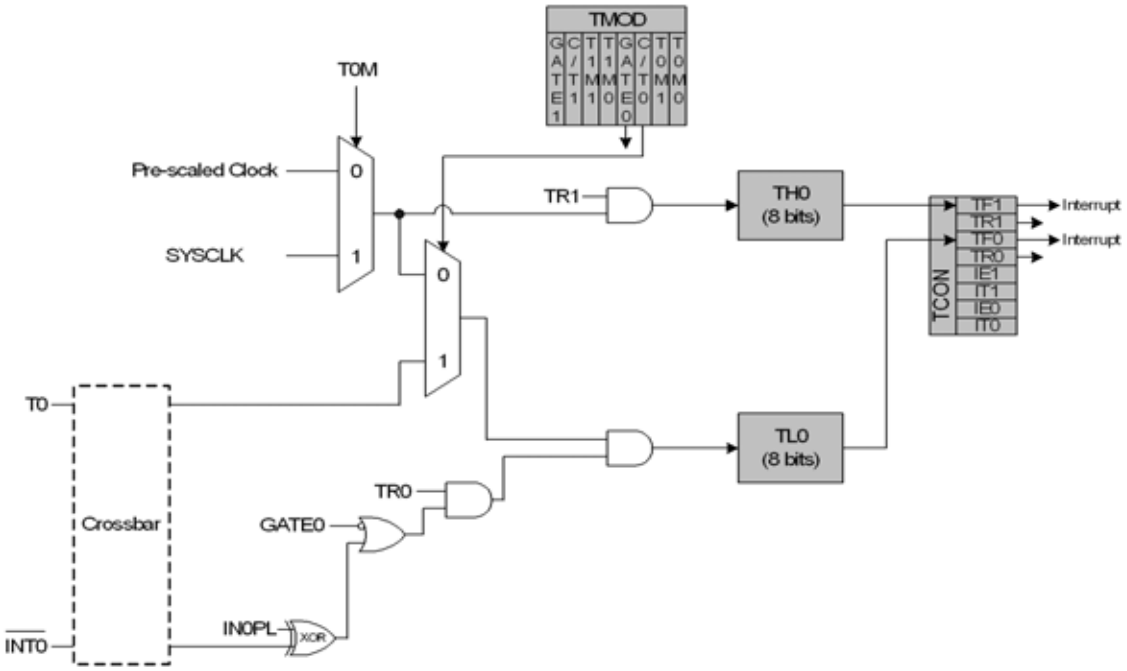


**Figure 25.3. T0 Mode 3 Block Diagram**

## 25.4. Timer 4

Timer 4 is a 16-bit timer formed by two 8-bit SFRs: TMR4L (low byte) and TMR4H (high byte). Timer 4 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T4SPLIT bit (TMR4CN.3) defines

Timer 4 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

### 25.4.1. 16-bit Timer with Auto-Reload

When T4SPLIT (TMR4CN.3) is zero, Timer 4 operates as a 16-bit timer with auto-reload. Timer 4 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 4 reload registers (TMR4RLH and TMR4RLL) is loaded into the Timer 4 register as shown in Figure 25.12, and the Timer 4 High Byte Overflow Flag (TMR4CN.7) is set. If Timer 4 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 4 overflow. Additionally, if Timer 4 interrupts are enabled and the TF4LEN bit is set (TMR4CN.5), an interrupt will be generated each time the lower 8 bits (TMR4L) overflow from 0xFF to 0x00.

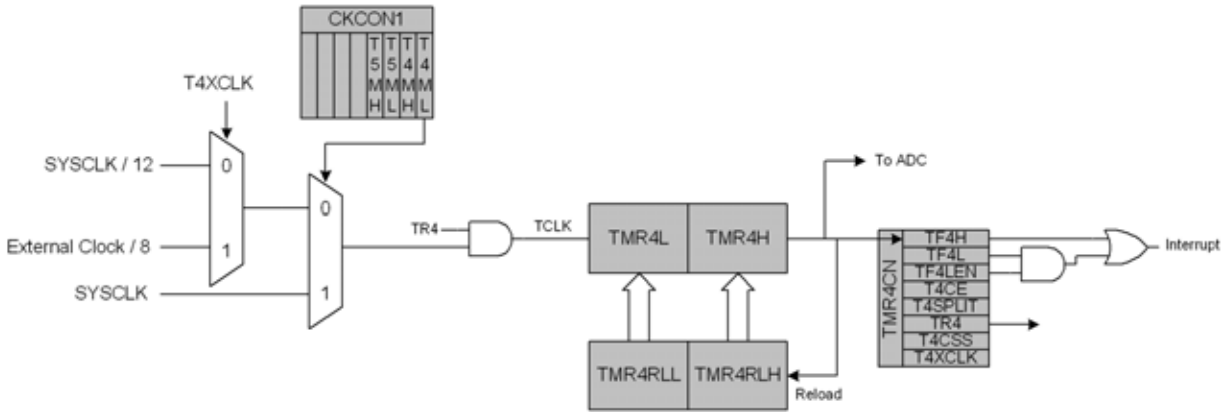


**Figure 25.12. Timer 4 16-Bit Mode Block Diagram**

## 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).

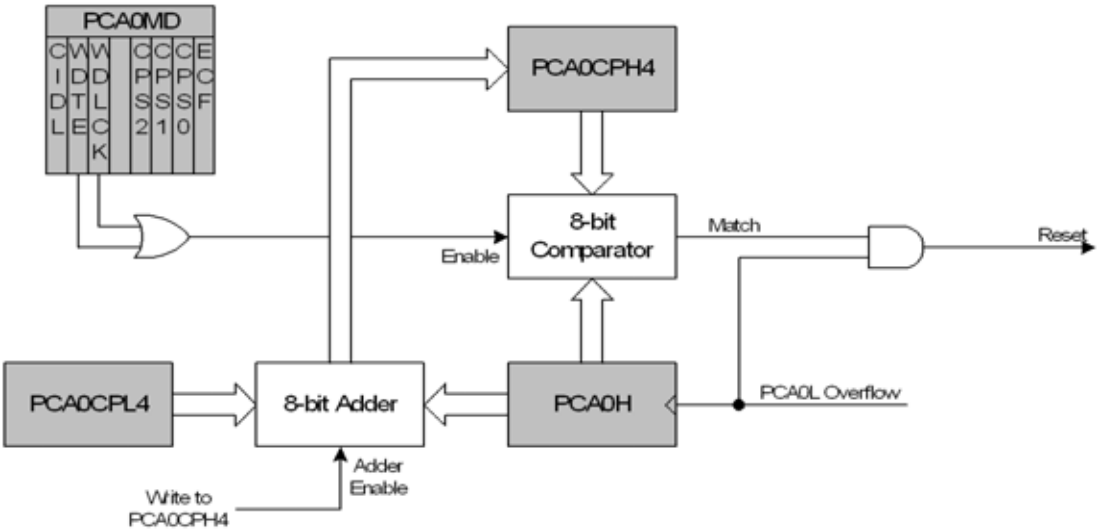


**Figure 26.10. PCA Module 4 with Watchdog Timer Enabled**

## 28.2. INT2 Pin Not Connected

### Problem

The INT2 pin is non-functional on revision A C8051F388/9/A/B devices.

### Impacts

The INT2 pin cannot be used to generate an interrupt.

### Workaround

The /INT0 and /INT1 external interrupts are available on P0 and can still be used to generate an external pin interrupt.

### Resolution

This behavior has been corrected on revision B of this device.