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 | Discontinued at Digi-Key |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 100MHz |
| Connectivity | EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 39 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | - |
| 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/c8051f363-gqr |

## 5.2. Temperature Sensor

The typical temperature sensor transfer function is shown in Figure 5.2. The output voltage ($V_{TEMP}$) is the positive ADC input when the temperature sensor is selected by bits AMX0P4-0 in register AMX0P.

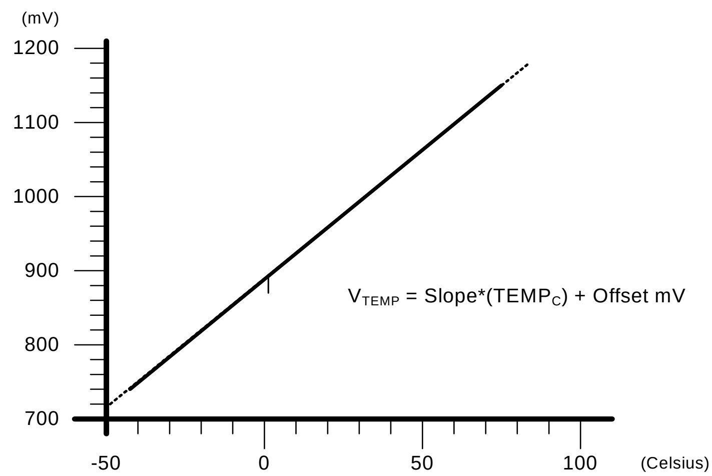


**Figure 5.2. Typical Temperature Sensor Transfer Function**

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.1 for linearity specifications). For absolute temperature measurements, gain and/ or offset calibration is recommended. Typically a 1-point calibration includes the following steps:

Step 1. Control/measure the ambient temperature (this temperature must be known).
Step 2. Power the device, and delay for a few seconds to allow for self-heating.
Step 3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
Step 4. Calculate the offset and/or gain characteristics, and store these values in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.3 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Note that parameters which affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

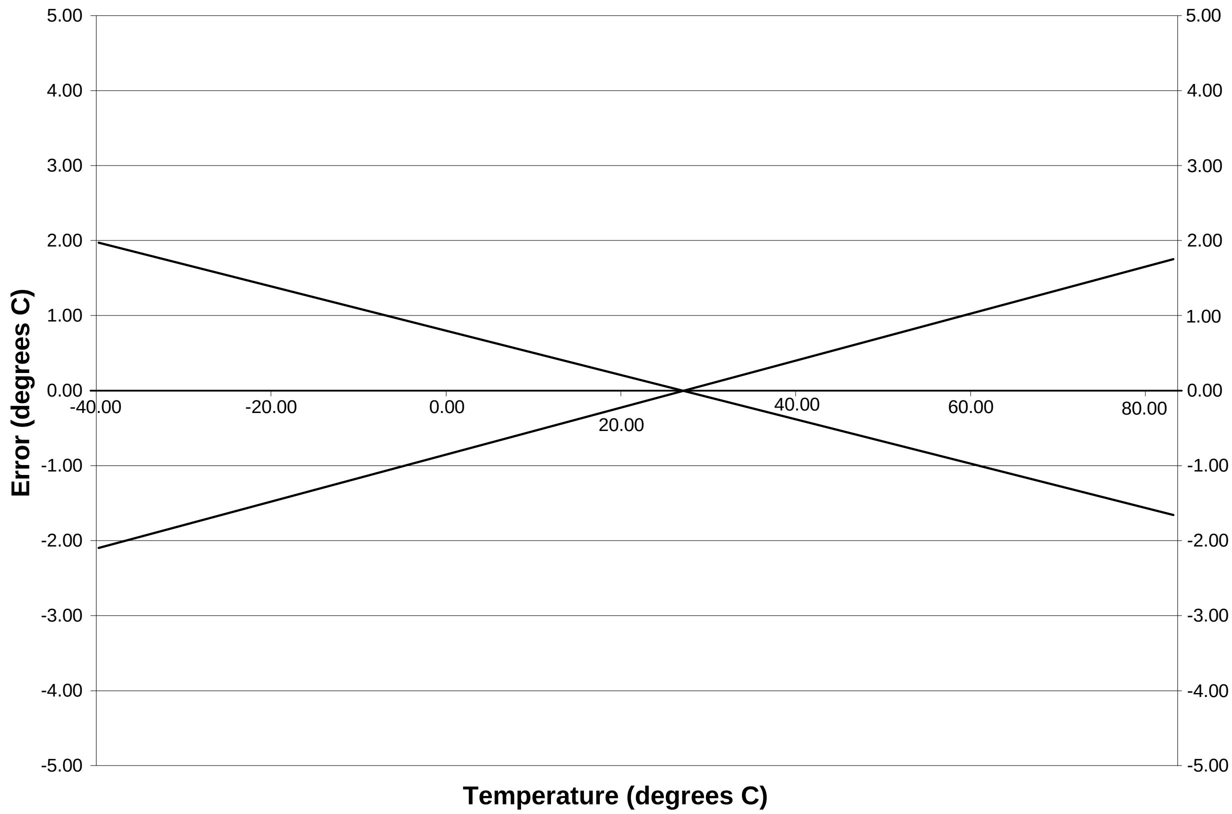


**Figure 5.3. Temperature Sensor Error with 1-Point Calibration**

## 5.4. Programmable Window Detector

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

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

**SFR Page: all pages**
**SFR Address: 0xC4**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–0: High byte of ADC0 Greater-Than Data Word.

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

**SFR Page: all pages**
**SFR Address: 0xC3**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–0: Low byte of ADC0 Greater-Than Data Word.

## Table 5.1. ADC0 Electrical Characteristics

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

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 10 | | bits |
| Integral Nonlinearity | | — | ±0.5 | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | ±0.5 | ±1 | LSB |
| Offset Error | | –12 | 3 | 12 | LSB |
| Full Scale Error | Differential mode | –5 | 1 | 5 | LSB |
| **Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)** | | | | | |
| Signal-to-Noise Plus Distortion | | 53 | 58 | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | –75 | — | dB |
| Spurious-Free Dynamic Range | | — | 75 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Conversion Clock | | — | — | 3 | MHz |
| Conversion Time in SAR Clocks | | 13 | — | — | clocks |
| Track/Hold Acquisition Time | | 300 | — | — | ns |
| Throughput Rate | | — | — | 200 | ksps |
| **Analog Inputs** | | | | | |
| ADC Input Voltage Range | Single Ended (AIN+ – GND)<br>Differential (AIN+ – AIN–) | 0<br>–VREF | —<br>— | VREF<br>VREF | V<br>V |
| Absolute Pin Voltage with respect to GND | Single Ended or Differential | 0 | — | $V_{DD}$ | V |
| Input Capacitance | | — | 5 | — | pF |
| **Temperature Sensor** | | | | | |
| Linearity* | | — | ±0.2 | — | °C |
| Slope | | — | 2.18 | — | mV/ºC |
| Slope Error* | | — | ±172 | — | µV/ºC |
| Offset | (Temp = 0 °C) | — | 802 | — | mV |
| Offset Error* | | — | ±18.5 | — | mV |
| **Power Specifications** | | | | | |
| Power Supply Current ($V_{DD}$ supplied to ADC0) | Operating Mode, 200 ksps | — | 450 | 900 | µA |
| Power Supply Rejection | | — | 3 | — | mV/V |

***Note:** Represents one standard deviation from the mean. Includes ADC offset, gain, and linearity variations.

complete Port I/O configuration details. The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data.

### SFR Definition 7.1. REF0CN: Reference Control

**SFR Page:** all pages
**SFR Address:** 0xD1

| R | R | R | R | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| – | – | – | – | REFSL | TEMPE | BIASE | REFBE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–4: UNUSED. Read = 0000b; Write = don't care.
Bit 3: REFSL: Voltage Reference Select.
This bit selects the source for the internal voltage reference.
0: VREF pin used as voltage reference.
1: $V_{DD}$ used as voltage reference.
Bit 2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.
Bit 1: BIASE: Internal Analog Bias Generator Enable Bit.
0: Internal Bias Generator off.
1: Internal Bias Generator on.
Bit 0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer disabled.
1: Internal Reference Buffer enabled. Internal voltage reference driven on the VREF pin.

## SFR Definition 10.4. EIP1: Extended Interrupt Priority 1

**SFR Page:** F
**SFR Address:** 0xCE

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PT3 | PCP1 | PCP0 | PPCA0 | PADC0 | PWADC0 | – | PSMB0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit 7: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupt.
0: Timer 3 interrupts set to low priority level.
1: Timer 3 interrupts set to high priority level.

Bit 6: PCP1: Comparator1 (CP1) Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 interrupt set to low priority level.
1: CP1 interrupt set to high priority level.

Bit 5: PCP0: Comparator0 (CP0) Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 interrupt set to low priority level.
1: CP0 interrupt set to high priority level.

Bit 4: PPCA0: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority level.
1: PCA0 interrupt set to high priority level.

Bit 3: PADC0: ADC0 Conversion Complete Interrupt Priority Control.
This bit sets the priority of the ADC0 Conversion Complete interrupt.
0: ADC0 Conversion Complete interrupt set to low priority level.
1: ADC0 Conversion Complete interrupt set to high priority level.

Bit 2: PWADC0: ADC0 Window Comparison Interrupt Priority Control.
This bit sets the priority of the ADC0 Window Comparison interrupt.
0: ADC0 Window Comparison interrupt set to low priority level.
1: ADC0 Window Comparison interrupt set to high priority level.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: PSMB0: SMBus (SMB0) Interrupt Priority Control.
This bit sets the priority of the SMB0 interrupt.
0: SMB0 interrupt set to low priority level.
1: SMB0 interrupt set to high priority level.

## SFR Definition 10.7. IT01CF: INT0/INT1 Configuration

**SFR Page:** all pages
**SFR Address:** 0xE4

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| IN1PL | IN1SL2 | IN1SL1 | IN1SL0 | IN0PL | IN0SL2 | IN0SL1 | IN0SL0 | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Note: Refer to SFR Definition 21.1. "TCON: Timer Control" on page 250 for INT0/1 edge- or level-sensitive interrupt selection.

Bit 7: IN1PL: /INT1 Polarity
0: /INT1 input is active low.
1: /INT1 input is active high.

Bits 6–4: IN1SL2-0: /INT1 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT1. Note that this pin assignment is independent of the Crossbar; /INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

| IN1SL2-0 | /INT1 Port Pin |
|---|---|
| 000 | P0.0 |
| 001 | P0.1 |
| 010 | P0.2 |
| 011 | P0.3 |
| 100 | P0.4 |
| 101 | P0.5 |
| 110 | P0.6 |
| 111 | P0.7 |

Bit 3: IN0PL: /INT0 Polarity
0: /INT0 interrupt is active low.
1: /INT0 interrupt is active high.

Bits 2–0: INT0SL2-0: /INT0 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT0. Note that this pin assignment is independent of the Crossbar. /INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

| IN0SL2-0 | /INT0 Port Pin |
|---|---|
| 000 | P0.0 |
| 001 | P0.1 |
| 010 | P0.2 |
| 011 | P0.3 |
| 100 | P0.4 |
| 101 | P0.5 |
| 110 | P0.6 |
| 111 | P0.7 |

## 12. 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. Refer to Section “16. Oscillators” on page 168 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source (Section “22.3. Watchdog Timer Mode” on page 270 details the use of the Watchdog Timer). Program execution begins at location 0x0000.

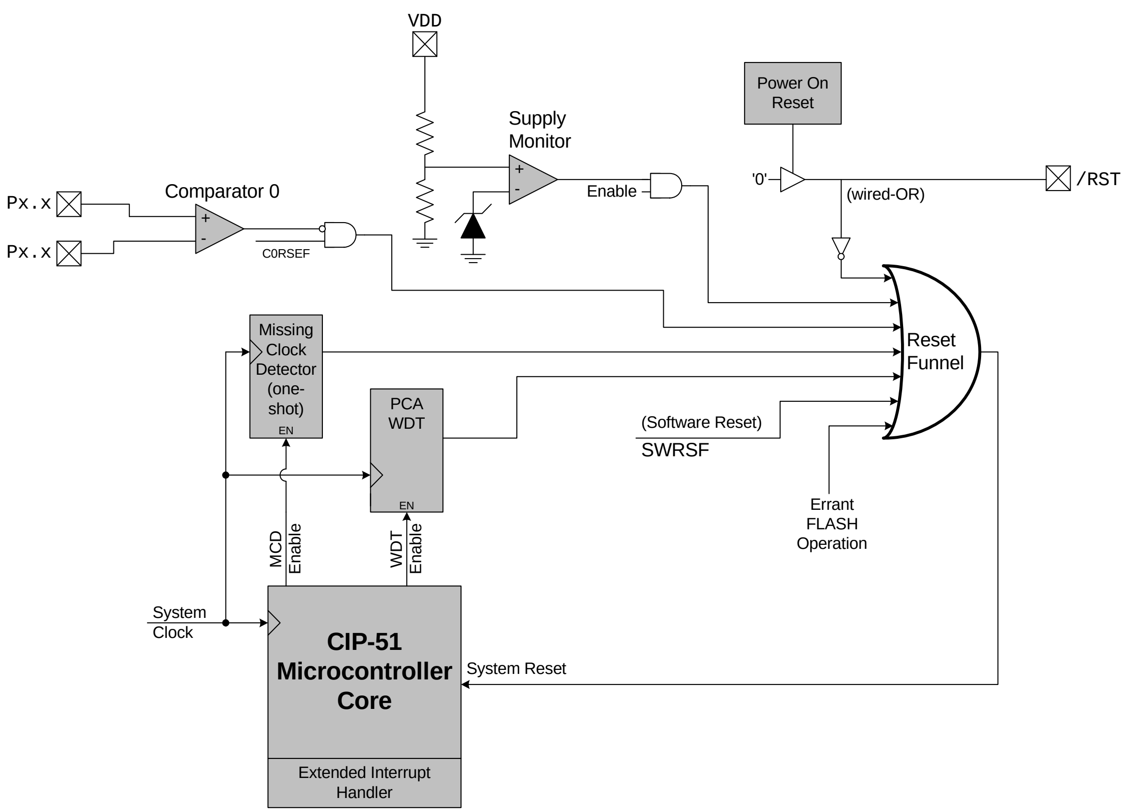


**Figure 12.1. Reset Sources**

# 13. Flash Memory

All devices include either 32 kB (C8051F360/1/2/3/4/5/6/7) or 16 kB (C8051F368/9) of on-chip, reprogrammable Flash memory for program code or non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface, or by software using the MOVX write instructions. Once cleared to logic ‘0’, a Flash bit must be erased to set it back to logic ‘1’. Bytes should be erased (set to 0xFF) before being reprogrammed. Flash write and erase operations are automatically timed by hardware for proper execution. During a Flash erase or write, the FLBUSY bit in the FLSTAT register is set to ‘1’ (see SFR Definition 14.5). During this time, instructions that are located in the prefetch buffer or the branch target cache can be executed, but the processor will stall until the erase or write is completed if instruction data must be fetched from Flash memory. Interrupts that have been pre-loaded into the branch target cache can also be serviced at this time, if the current code is also executing from the prefetch engine or cache memory. Any interrupts that are not pre-loaded into cache, or that occur while the core is halted, will be held in a pending state during the Flash write/erase operation, and serviced in priority order once the Flash operation has completed. Refer to Table 13.2 for the electrical characteristics of the Flash memory.

## 13.1. Programming the Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “24. C2 Interface” on page 283. For detailed guidelines on writing or erasing Flash from firmware, please see Section “13.3. Flash Write and Erase Guidelines” on page 140.

The Flash memory can be programmed from software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic ‘1’. This directs the MOVX writes to Flash memory instead of to XRAM, which is the default target. The PSWE bit remains set until cleared by software. To avoid errant Flash writes, it is recommended that interrupts be disabled while the PSWE bit is logic ‘1’.

Flash memory is read using the MOVC instruction. MOVX reads are always directed to XRAM, regardless of the state of PSWE.

**Note: To ensure the integrity of the Flash contents, the on-chip $V_{DD}$ Monitor must be enabled in any system that includes code that writes and/or erases Flash memory from software. Furthermore, there should be no delay between enabling the $V_{DD}$ Monitor and enabling the $V_{DD}$ Monitor as a reset source. Any attempt to write or erase Flash memory while the $V_{DD}$ Monitor disabled will cause a Flash Error device reset.**

A write to Flash memory can clear bits but cannot set them; only an erase operation can set bits in Flash. **A byte location to be programmed must be erased before a new value can be written**.

Write/Erase timing is automatically controlled by hardware. Note that on the 32 k Flash devices, 1024 bytes beginning at location 0x7C00 are reserved. Flash writes and erases targeting the reserved area should be avoided.

### 13.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and

### 13.3.2. 16.4.2 PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a '1'. There should be exactly one routine in code that sets PSWE to a '1' to write Flash bytes and one routine in code that sets both PSWE and PSEE both to a '1' to erase Flash pages.
8. Minimize the number of variable accesses while PSWE is set to a '1'. Handle pointer address updates and loop maintenance outside the "PSWE = 1; ... PSWE = 0;" area. Code examples showing this can be found in AN201, "Writing to Flash from Firmware", available from the Silicon Laboratories web site.
9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to '0'. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

### 13.3.3. System Clock

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

# 14. Branch Target Cache

The C8051F36x device families incorporate a 32x4 byte branch target cache with a 4-byte prefetch engine. Because the access time of the Flash memory is 40 ns, and the minimum instruction time is 10 ns (C8051F360/1/2/3/4/5/6/7) or 20 ns (C8051F368/9), the branch target cache and prefetch engine are necessary for full-speed code execution. Instructions are read from Flash memory four bytes at a time by the prefetch engine, and given to the CIP-51 processor core to execute. When running linear code (code without any jumps or branches), the prefetch engine alone allows instructions to be executed at full speed. When a code branch occurs, a search is performed for the branch target (destination address) in the cache. If the branch target information is found in the cache (called a “cache hit”), the instruction data is read from the cache and immediately returned to the CIP-51 with no delay in code execution. If the branch target is not found in the cache (called a “cache miss”), the processor may be stalled for up to four clock cycles while the next set of four instructions is retrieved from Flash memory. Each time a cache miss occurs, the requested instruction data is written to the cache if allowed by the current cache settings. A data flow diagram of the interaction between the CIP-51 and the Branch Target Cache and Prefetch Engine is shown in Figure 14.1.

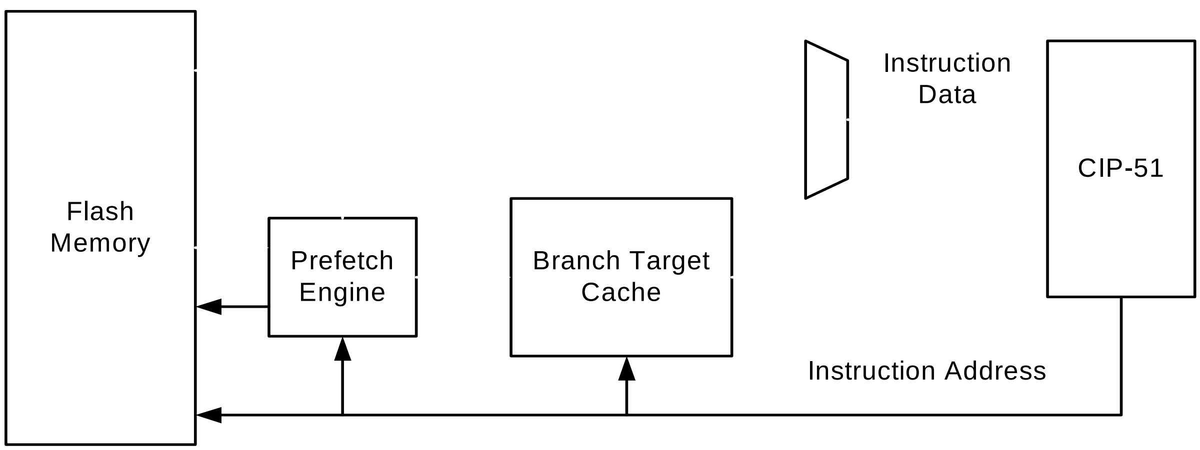


**Figure 14.1. Branch Target Cache Data Flow**

## 14.1. Cache and Prefetch Operation

The branch target cache maintains two sets of memory locations: “slots” and “tags”. A slot is where the cached instruction data from Flash is stored. Each slot holds four consecutive code bytes. A tag contains the 13 most significant bits of the corresponding Flash address for each four-byte slot. Thus, instruction data is always cached along four-byte boundaries in code space. A tag also contains a “valid bit”, which indicates whether a cache location contains valid instruction data. A special cache location (called the linear tag and slot), is reserved for use by the prefetch engine. The cache organization is shown in Figure 14.2. Each time a Flash read is requested, the address is compared with all valid cache tag locations (including the linear tag). If any of the tag locations match the requested address, the data from that slot is immediately provided to the CIP-51. If the requested address matches a location that is currently being read by the prefetch engine, the CIP-51 will be stalled until the read is complete. If a match is not found, the current prefetch operation is abandoned, and a new prefetch operation is initiated for the requested instruction data. When the prefetch operation is finished, the CIP-51 begins executing the instructions that were retrieved, and the prefetch engine begins reading the next four-byte word from Flash memory. If the newly-fetched data also meets the criteria necessary to be cached, it will be written to the cache in the slot indicated by the current replacement algorithm.

### 15.6.2. Multiplexed Mode

#### 15.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

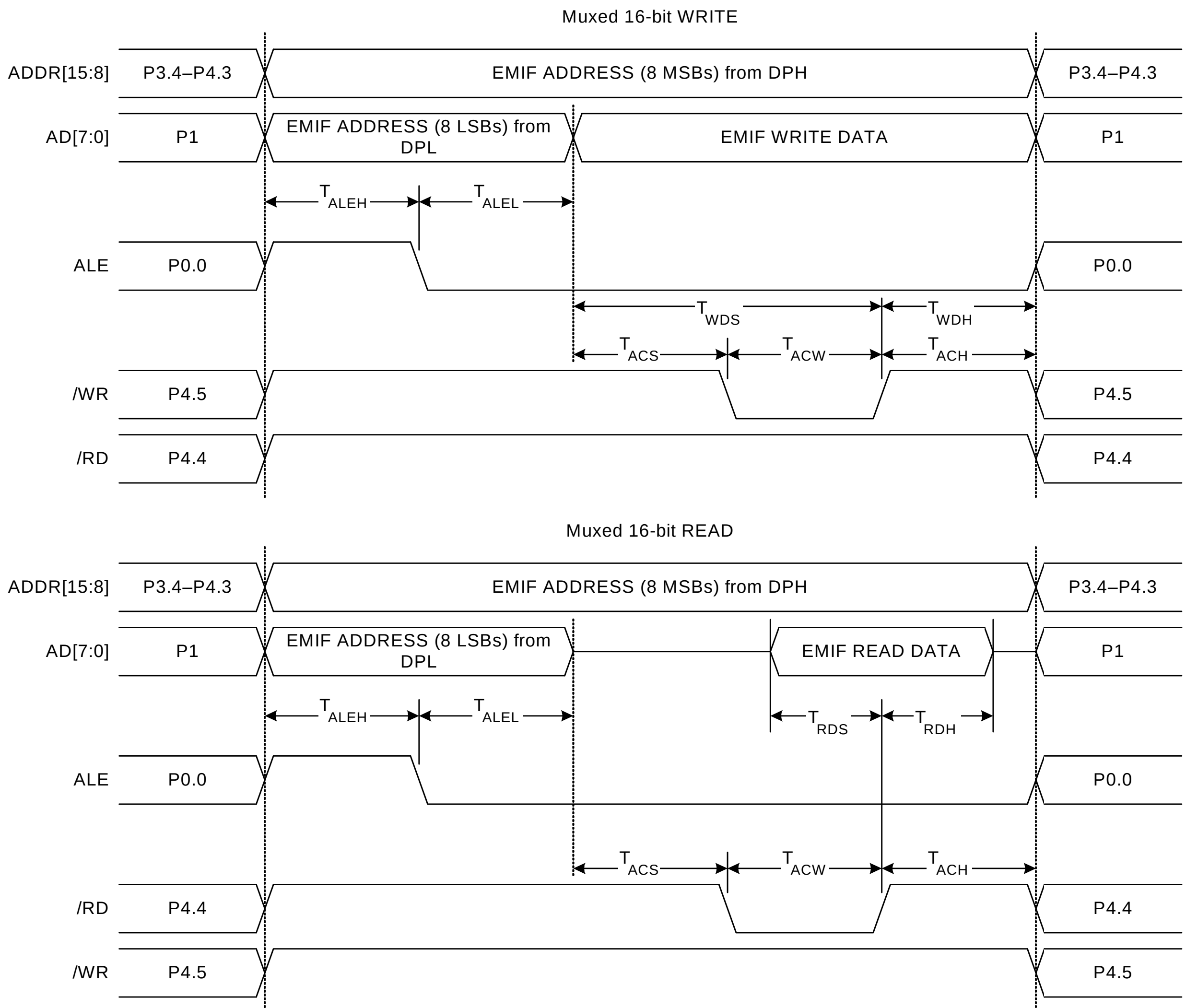


**Figure 15.7. Multiplexed 16-bit MOVX Timing**

## SFR Definition 16.2. OSCICN: Internal Oscillator Control

**SFR Page:** F
**SFR Address:** 0xB7

| R/W | R | R/W | R | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| IOSCEN | IFRDY | SUSPEND | Reserved | Reserved | Reserved | IFCN1 | IFCN0 | 11000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit 7: IOSCEN: Internal Oscillator Enable Bit.
0: Internal Oscillator Disabled.
1: Internal Oscillator Enabled.

Bit 6: IFRDY: Internal Oscillator Frequency Ready Flag.
0: Internal Oscillator not running at programmed frequency.
1: Internal Oscillator running at programmed frequency.

Bits 5: SUSPEND: Internal Oscillator Suspend Enable Bit.
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 occur.

Bits 4–2: RESERVED. Read = 000b. Must Write 000b.

Bits 1–0: IFCN1-0: Internal Oscillator Frequency Control Bits.
00: Internal Oscillator is divided by 8. (default)
01: Internal Oscillator is divided by 4.
10: Internal Oscillator is divided by 2.
11: Internal Oscillator is divided by 1.

## Table 16.1. Internal High Frequency Oscillator Electrical Characteristics

–40°C to +85°C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Calibrated Internal Oscillator Frequency | | 24 | 24.5 | 25 | MHz |
| Internal Oscillator Supply Current (from $V_{DD}$) | OSCICN.7 = 1 | — | 450 | 600 | µA |
| Power Supply Sensitivity | Constant Temperature | — | 0.12 | — | %/V |
| Temperature Sensitivity | Constant Supply | — | 60 | — | ppm/°C |
| External Clock Frequency | | 0 | — | 30 | MHz |
| $T_{XCH}$ (External Clock High Time) | | 15 | — | — | ns |
| $T_{XCL}$ (External Clock Low Time) | | 15 | — | — | ns |

## 16.2. Programmable Internal Low-Frequency (L-F) Oscillator

All C8051F36x devices include a programmable low-frequency internal oscillator, which is calibrated to a nominal frequency of 80 kHz. The low-frequency oscillator circuit includes a divider that can be changed to divide the clock by 1, 2, 4, or 8, using the OSCLD bits in the OSCLCN register (see SFR Definition 16.3). Additionally, the OSCLF bits (OSCLCN5:2) can be used to adjust the oscillator's output frequency.

## 16.5. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 16.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 16.5 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is enabled, the oscillator amplitude detection circuit requires a settle time to achieve proper bias. Waiting at least 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Force the XTAL1 and XTAL2 pins low by writing 0's to the port latch.
Step 2. Configure XTAL1 and XTAL2 as analog inputs.
Step 3. Enable the external oscillator.
Step 4. Wait at least 1 ms.
Step 5. Poll for XTLVLD => '1'.
Step 6. Switch the system clock to the external oscillator.

**Note:** Tuning-fork crystals may require additional settling time before XTLVLD returns a valid result.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins.

**Note:** The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 16.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 16.2.

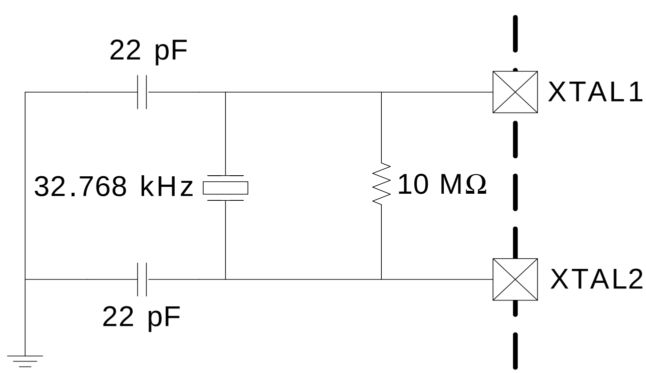


**Figure 16.2. 32.768 kHz External Crystal Example**

**Important Note on External Crystals:** Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

## 16.6. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 16.1, Option 2. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation. If the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

$f = 1.23(10^3)/RC = 1.23\ (10^3)/[246 \times 50] = 0.1\ MHz = 100\ kHz$

Referring to the table in SFR Definition 16.5, the required XFCN setting is 010b. Programming XFCN to a higher setting in RC mode will improve frequency accuracy at a slightly increased external oscillator supply current.

## 16.7. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 16.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation from the equations below. Assume $V_{DD}$ = 3.0 V and f = 75 kHz:

$f = KF / (C \times V_{DD})$
$0.075\ MHz = KF / (C \times 3.0)$

Since the frequency of roughly 75 kHz is desired, select the K Factor from the table in SFR Definition 16.5 as KF = 7.7:

$0.075\ MHz = 7.7 / (C \times 3.0)$

$C \times 3.0 = 7.7 / 0.075\ MHz$

$C = 102.6 / 3.0\ pF = 34.2\ pF$

Therefore, the XFCN value to use in this example is 010b.

## 17.3. General Purpose Port I/O

Port pins that remain unassigned by the Crossbar and are not used by analog peripherals can be used for general purpose I/O. Ports P0-P3 are accessed through corresponding special function registers (SFRs) that are both byte-addressable and bit-addressable. Port 4 (C8051F360/3 only) uses an SFR which is byte-addressable. When writing to a Port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the Port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the Crossbar, the Port register can always read its corresponding Port I/O pin). The exception to this is the execution of the read-modify-write instructions that target a Port Latch register as the destination. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a Port SFR. For these instructions, the value of the latch register (not the pin) is read, modified, and written back to the SFR.

In addition to performing general purpose I/O, P0, P1, and P2 can generate a port match event if the logic levels of the Port's input pins match a software controlled value. A port match event is generated if (P0 & P0MASK) does not equal (P0MATCH & P0MASK), if (P1 & P1MASK) does not equal (P1MATCH & P1MASK), or if (P2 & P2MASK) does not equal (P2MATCH & P2MASK). This allows Software to be notified if a certain change or pattern occurs on P0, P1, or P2 input pins regardless of the XBRn settings. A port match event can cause an interrupt if EMAT (EIE2.1) is set to '1' or cause the internal oscillator to awaken from SUSPEND mode. See Section “16.1.1. Internal Oscillator Suspend Mode” on page 169 for more information.

### SFR Definition 17.3. P0: Port0

**SFR Page:** all pages
**SFR Address:** 0x80 (bit addressable)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–0: P0.[7:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P0MDOUT.n bit = 0).
Read - Always reads ‘0’ if selected as analog input in register P0MDIN. Directly reads Port pin when configured as digital input.
0: P0.n pin is logic low.
1: P0.n pin is logic high.

## 19.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown below.

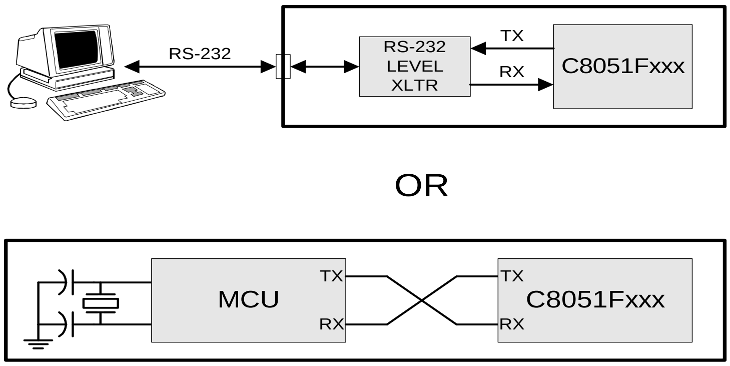


**Figure 19.3. UART Interconnect Diagram**

### 19.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic ‘1’. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic ‘0’, and if MCE0 is logic ‘1’, the stop bit must be logic ‘1’. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

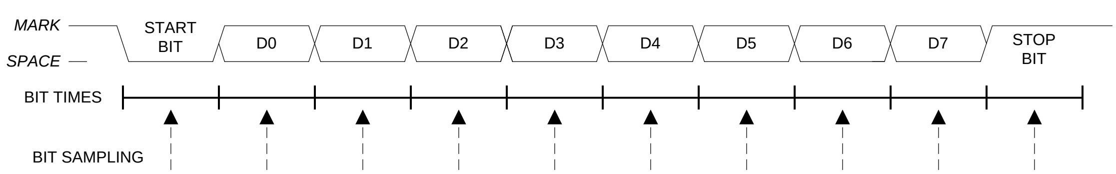


**Figure 19.4. 8-Bit UART Timing Diagram**

## SFR Definition 21.9. TMR2RLL: Timer 2 Reload Register Low Byte

**SFR Page:** all pages
**SFR Address:** 0xCA

| 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 | |

Bits 7–0: TMR2RLL: Timer 2 Reload Register Low Byte.
TMR2RLL holds the low byte of the reload value for Timer 2.

## SFR Definition 21.10. TMR2RLH: Timer 2 Reload Register High Byte

**SFR Page:** all pages
**SFR Address:** 0xCB

| 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 | |

Bits 7–0: TMR2RLH: Timer 2 Reload Register High Byte.
The TMR2RLH holds the high byte of the reload value for Timer 2.

## SFR Definition 21.11. TMR2L: Timer 2 Low Byte

**SFR Page:** all pages
**SFR Address:** 0xCC

| 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 | |

Bits 7–0: TMR2L: Timer 2 Low Byte.
In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

## SFR Definition 21.12. TMR2H Timer 2 High Byte

**SFR Page:** all pages
**SFR Address:** 0xCD

| 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 | |

Bits 7–0: TMR2H: Timer 2 High Byte.
In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

## 22.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter. Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2–CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 22.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic ‘1’ and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic ‘1’ enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic ‘1’). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

**Table 22.1. PCA Timebase Input Options**

| CPS2 | CPS1 | CPS0 | Timebase |
|---|---|---|---|
| 0 | 0 | 0 | System clock divided by 12 |
| 0 | 0 | 1 | System clock divided by 4 |
| 0 | 1 | 0 | Timer 0 overflow |
| 0 | 1 | 1 | High-to-low transitions on ECI (max rate = system clock divided by 4) |
| 1 | 0 | 0 | System clock |
| 1 | 0 | 1 | External oscillator source divided by 8* |
| 1 | 1 | 0 | RESERVED |
| 1 | 1 | 1 | RESERVED |

***Note:** External clock divided by 8 is synchronized with the system clock.

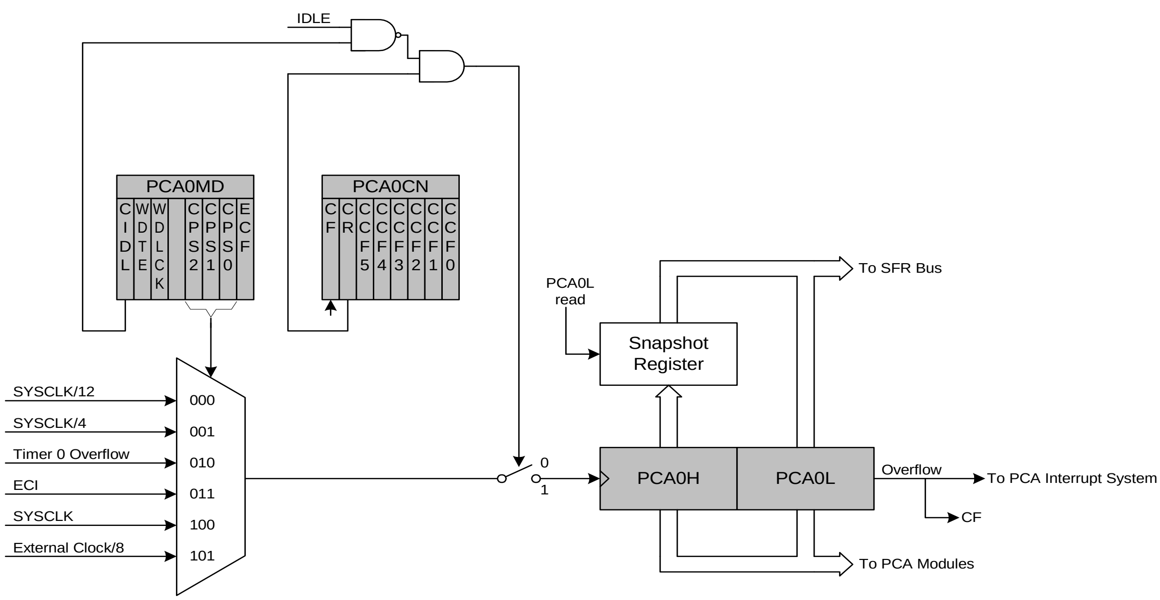


**Figure 22.2. PCA Counter/Timer Block Diagram**

### 22.2.5. 8-Bit Pulse Width Modulator Mode

Each module can be used independently to generate pulse width modulated (PWM) outputs on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA0 counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA0 counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be high. When the count value in PCA0L overflows, the CEXn output will be low (see Figure 22.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the counter/timer's high byte (PCA0H) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register enables 8-Bit Pulse Width Modulator mode. The duty cycle for 8-Bit PWM Mode is given by Equation 22.2.

**Equation 22.2. 8-Bit PWM Duty Cycle**

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Using Equation 22.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to ‘0’.

**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’.

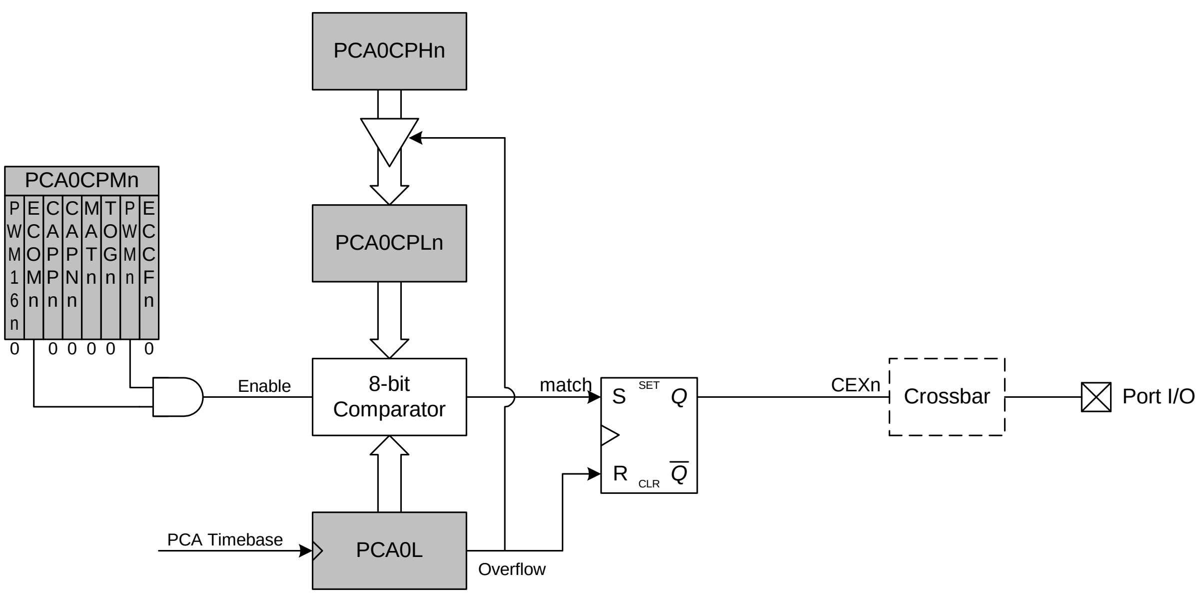


**Figure 22.8. PCA 8-Bit PWM Mode Diagram**