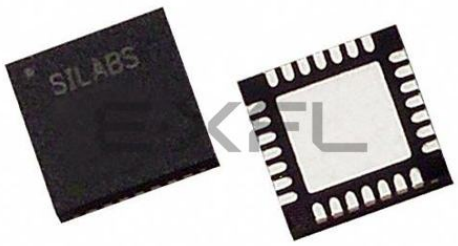



Welcome to E-XFL.COM

## 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 | 50MHz |
| Connectivity | SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 25 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 17x10b; D/A 1x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-VFQFN Exposed Pad |
| Supplier Device Package | 28-QFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f369-gm |

## List of Tables

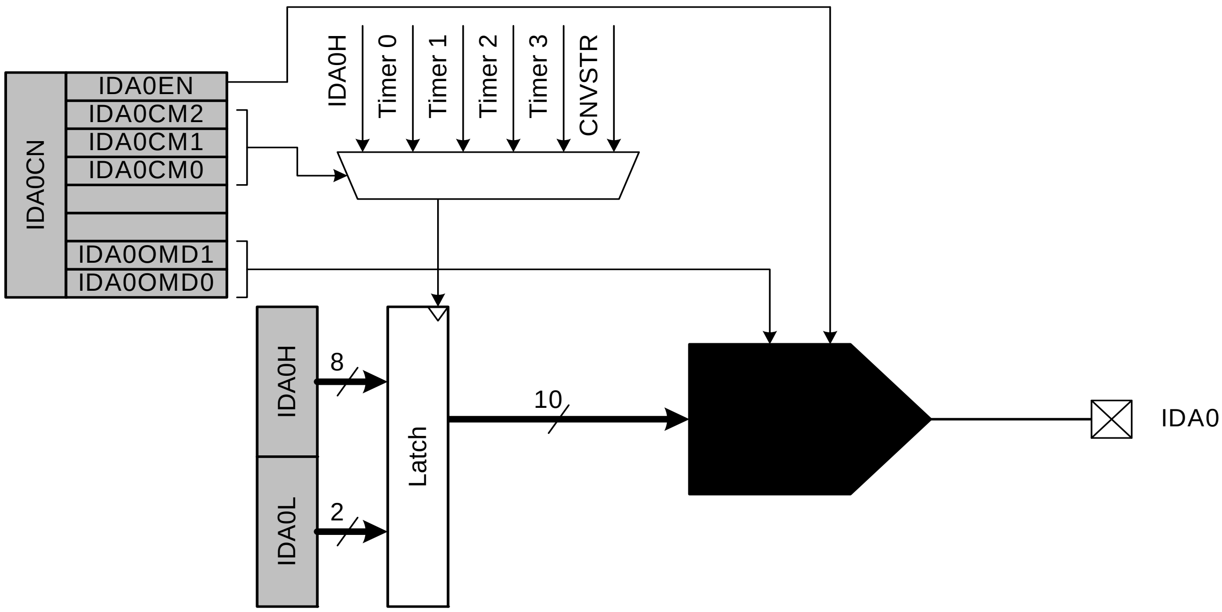



**Figure 1.14. IDA0 Functional Block Diagram**

# 3. Global Electrical Characteristics

## Table 3.1. Global Electrical Characteristics

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

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Digital Supply Voltage | SYSCLK = 0 to 50 MHz<br>SYSCLK > 50 MHz | 2.7<br>3.0 | 3.0<br>3.3 | 3.6<br>3.6 | V |
| Digital Supply RAM Data Retention Voltage | | — | 1.5 | — | V |
| SYSCLK (System Clock)[1,2] | C8051F360/1/2/3/4/5<br>C8051F366/7/8/9 | 0<br>0 | —<br>— | 100<br>50 | MHz<br>MHz |
| Specified Operating Temperature Range | | –40 | — | +85 | °C |
| **Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)** | | | | | |
| $I_{DD}$[2] | $V_{DD}$ = 3.6 V, F = 100 MHz | — | 68 | 75 | mA |
| | $V_{DD}$ = 3.6 V, F = 25 MHz | — | 21 | 25 | mA |
| | $V_{DD}$ = 3.0 V, F = 100 MHz | — | 54 | 60 | mA |
| | $V_{DD}$ = 3.0 V, F = 25 MHz | — | 16 | 18 | mA |
| | $V_{DD}$ = 3.0 V, F = 1 MHz | — | 0.48 | — | mA |
| | $V_{DD}$ = 3.0 V, F = 80 kHz | — | 36 | — | µA |
| $I_{DD}$ Supply Sensitivity[3] | F = 25 MHz | — | 56 | — | %/V |
| | F = 1 MHz | — | 57 | — | %/V |
| $I_{DD}$ Frequency Sensitivity[3,4] | $V_{DD}$ = 3.0 V, F <= 20 MHz, T = 25 °C | — | 0.45 | — | mA/MHz |
| | $V_{DD}$ = 3.0 V, F > 20 MHz, T = 25 °C | — | 0.38 | — | mA/MHz |
| | $V_{DD}$ = 3.6 V, F <= 20 MHz, T = 25 °C | — | 0.61 | — | mA/MHz |
| | $V_{DD}$ = 3.6 V, F > 20 MHz, T = 25 °C | — | 0.51 | — | mA/MHz |

## 5.3. Modes of Operation

ADC0 has a maximum conversion speed of 200 ksps. The ADC0 conversion clock is a divided version of the system clock, determined by the AD0SC bits in the ADC0CF register (system clock divided by (AD0SC + 1) for $0 \leq AD0SC \leq 31$).

### 5.3.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2-0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a ‘1’ to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 1 overflow
5. A rising edge on the CNVSTR input signal
6. A Timer 3 overflow

Writing a ‘1’ to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic ‘1’ and reset to logic ‘0’ when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic ‘1’. Note that when Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See Section “21. Timers” on page 245 for timer configuration.

**Important Note About Using CNVSTR:** The CNVSTR input pin also functions as Port pin P0.7 on the C8051F360 devices and Port pin P0.6 on the C8051F361/2/6/7/8/9 devices. When the CNVSTR input is used as the ADC0 conversion source, the corresponding port pin should be skipped by the Digital Crossbar. To configure the Crossbar to skip the port pin, set the appropriate bit to ‘1’ in register P0SKIP. See Section “17. Port Input/Output” on page 182 for details on Port I/O configuration.

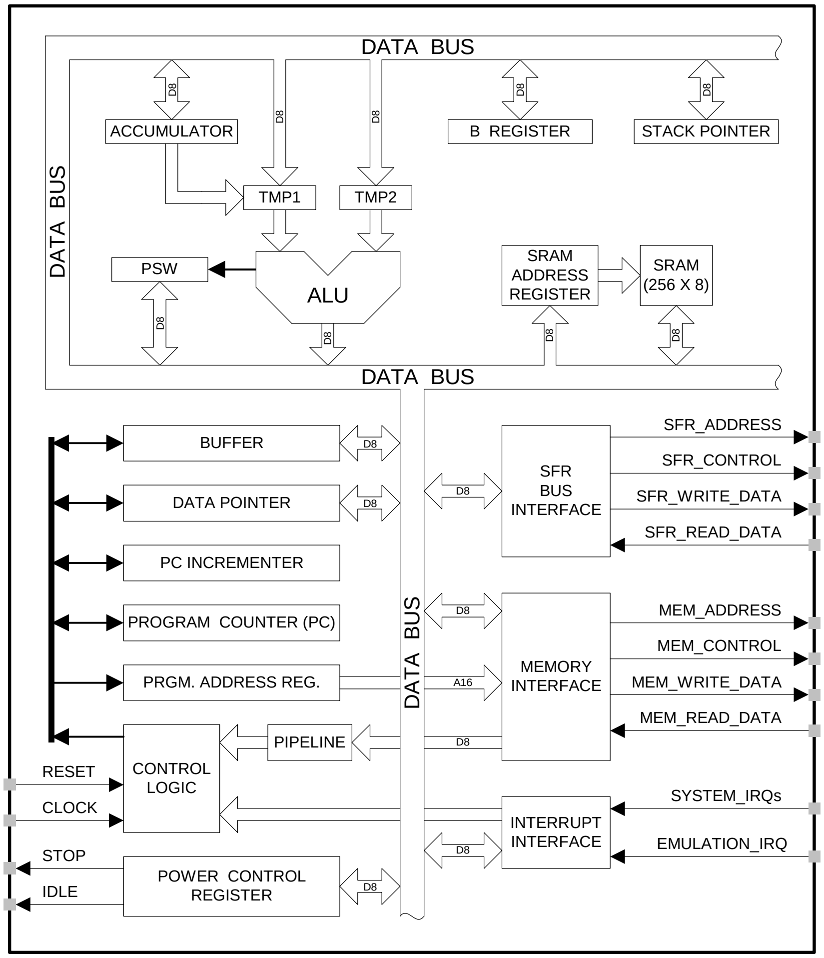



**Figure 9.1. CIP-51 Block Diagram**

## 9.2. Programming and Debugging Support

A C2-based serial interface is provided for in-system programming of the Flash program memory and communication with on-chip debug support logic. The re-programmable Flash can also be read and changed by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints and watch points, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debug is completely non-intrusive and non-invasive, requiring no RAM, Stack, timers, or other on-chip resources.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, macro assembler, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via its C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

## 9.4. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. There are 256 bytes of internal data memory and 32k bytes (C8051F360/1/2/3/4/5/6/7) or 16k bytes (C8051F368/9) of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 9.2.

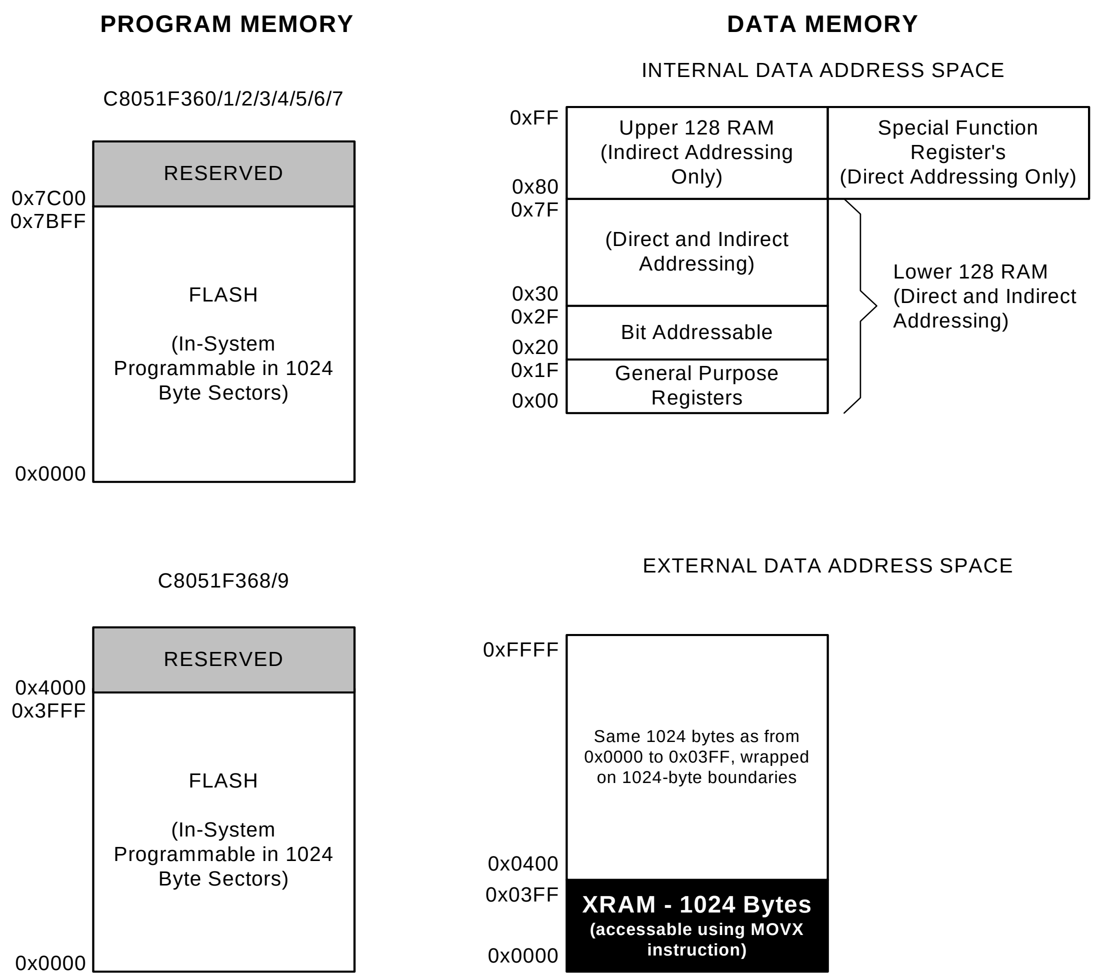



**Figure 9.2. Memory Map**

### 9.4.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F360/1/2/3/4/5/6/7 implement 32 kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x7BFF. Addresses above 0x7BFF are reserved on the 32 kB devices. The C8051F368/9 implement 16 kB of Flash from addresses 0x0000 to 0x3FFF.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section “13. Flash Memory” on page 135 for further details.

## SFR Definition 11.1. MAC0CF: MAC0 Configuration

**SFR Page: 0**
**SFR Address: 0xD7**

| R | R | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| – | – | MAC0SC | MAC0SD | MAC0CA | MAC0SAT | MAC0FM | MAC0MS | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–6: UNUSED: Read = 00b, Write = don't care.
Bit 5: MAC0SC: Accumulator Shift Control.
When set to 1, the 40-bit MAC0 Accumulator register will be shifted during the next SYSCLK cycle. The direction of the shift (left or right) is controlled by the MAC0SD bit.
This bit is cleared to '0' by hardware when the shift is complete.
Bit 4: MAC0SD: Accumulator Shift Direction.
This bit controls the direction of the accumulator shift activated by the MAC0SC bit.
0: MAC0 Accumulator will be shifted left.
1: MAC0 Accumulator will be shifted right.
Bit 3: MAC0CA: Clear Accumulator.
This bit is used to reset MAC0 before the next operation.
When set to '1', the MAC0 Accumulator will be cleared to zero and the MAC0 Status register will be reset during the next SYSCLK cycle.
This bit will be cleared to '0' by hardware when the reset is complete.
Bit 2: MAC0SAT: Saturate Rounding Register.
This bit controls whether the Rounding Register will saturate. If this bit is set and a Soft Overflow occurs, the Rounding Register will saturate. This bit does not affect the operation of the MAC0 Accumulator. See Section 11.6 for more details about rounding and saturation.
0: Rounding Register will not saturate.
1: Rounding Register will saturate.
Bit 1: MAC0FM: Fractional Mode.
This bit selects between Integer Mode and Fractional Mode for MAC0 operations.
0: MAC0 operates in Integer Mode.
1: MAC0 operates in Fractional Mode.
Bit 0: MAC0MS: Mode Select
This bit selects between MAC Mode and Multiply Only Mode.
0: MAC (Multiply and Accumulate) Mode.
1: Multiply Only Mode.

**Note:**The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 11.7. MAC0ACC3: MAC0 Accumulator Byte 3

**SFR Page: 0**
**SFR Address: 0xD5**

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–0: Byte 3 (bits 31–24) of MAC0 Accumulator.

**Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 11.8. MAC0ACC2: MAC0 Accumulator Byte 2

**SFR Page: 0**
**SFR Address: 0xD4**

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–0: Byte 2 (bits 23–16) of MAC0 Accumulator.

**Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 11.9. MAC0ACC1: MAC0 Accumulator Byte 1

**SFR Page: 0**
**SFR Address: 0xD3**

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7–0: Byte 1 (bits 15–8) of MAC0 Accumulator.

**Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

### 15.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = '101' or '111'.

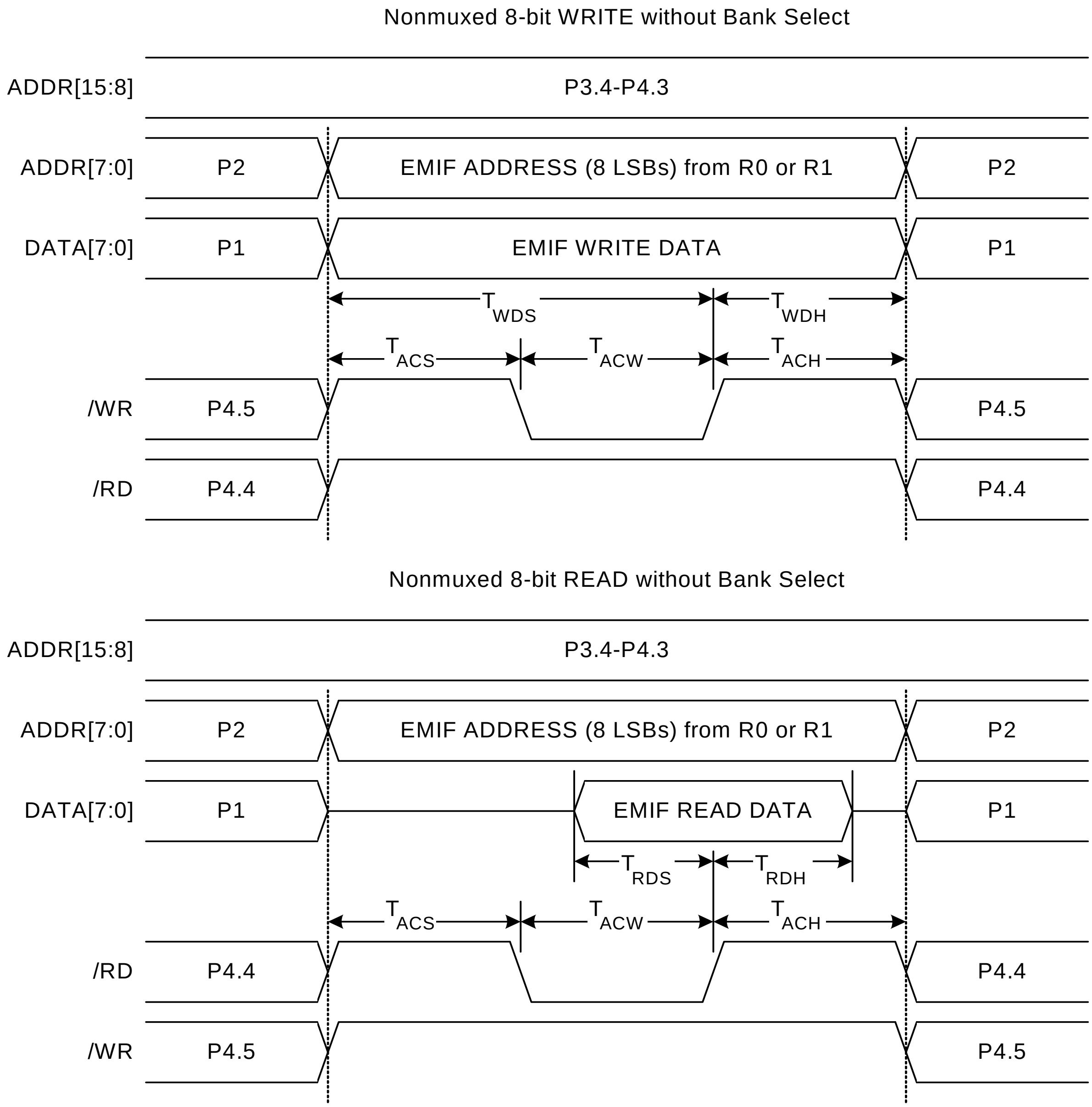
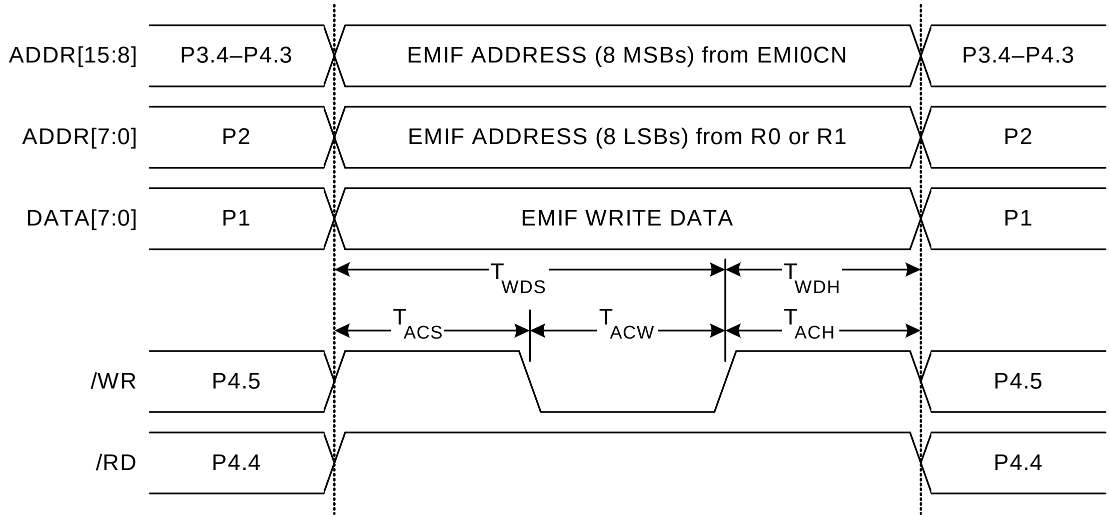



**Figure 15.5. Non-multiplexed 8-bit MOVX without Bank Select Timing**

**15.6.1.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘110’.**

Nonmuxed 8-bit WRITE with Bank Select




Nonmuxed 8-bit READ with Bank Select

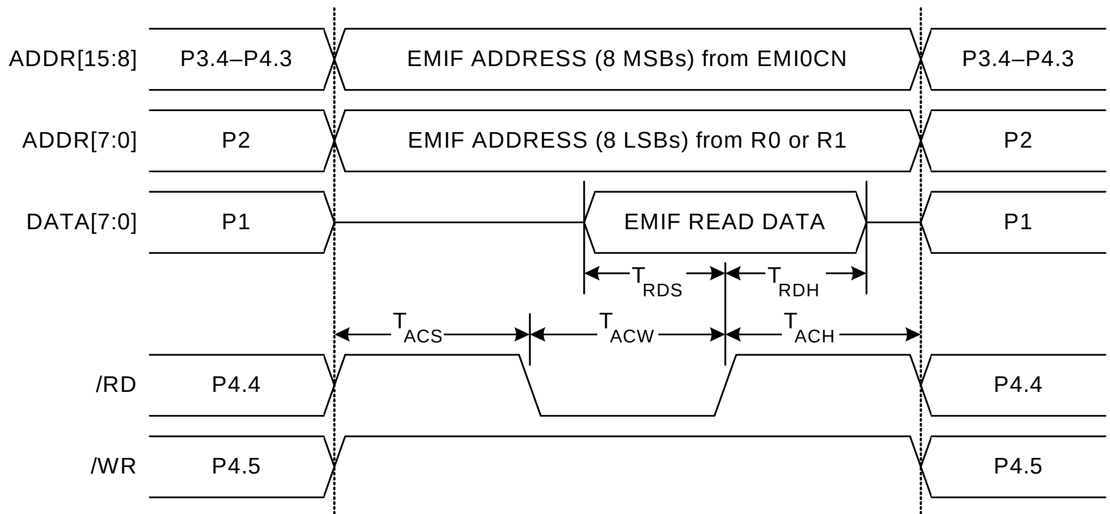



**Figure 15.6. Non-multiplexed 8-bit MOVX with Bank Select Timing**

## 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.

## 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.

## SFR Definition 17.18. P2SKIP: Port2 Skip

**SFR Page:** F
**SFR Address:** 0xD6

| 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: P2SKIP[7:0]: Port2 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions ($V_{REF}$ input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P2.n pin is not skipped by the Crossbar.
1: Corresponding P2.n pin is skipped by the Crossbar.

## SFR Definition 17.19. P2MAT: Port2 Match

**SFR Page:** 0
**SFR Address:** 0xB1

| 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: P2MAT[7:0]: Port2 Match Value.
These bits control the value that unmasked P2 Port pins are compared against. A Port Match event is generated if (P2 & P2MASK) does not equal (P2MAT & P2MASK).

## SFR Definition 17.20. P2MASK: Port2 Mask

**SFR Page:** 0
**SFR Address:** 0xB2

| 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: P2MASK[7:0]: Port2 Mask Value.
These bits select which Port pins will be compared to the value stored in P2MAT.
0: Corresponding P2.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P2.n pin is compared to the corresponding bit in P2MAT.

Figure 18.4 shows the typical SCL generation described by Equation 18.2. Notice that $T_{HIGH}$ is typically twice as large as $T_{LOW}$. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 18.1.

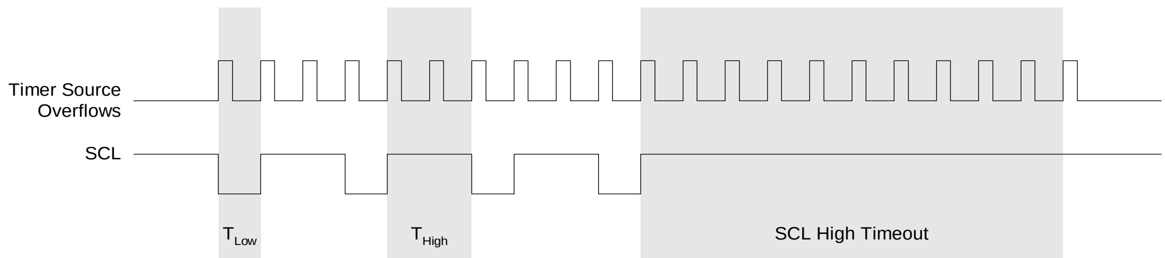



**Figure 18.4. Typical SMBus SCL Generation**

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 18.2 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 18.2. 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 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “18.3.3. SCL Low Timeout” on page 202). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

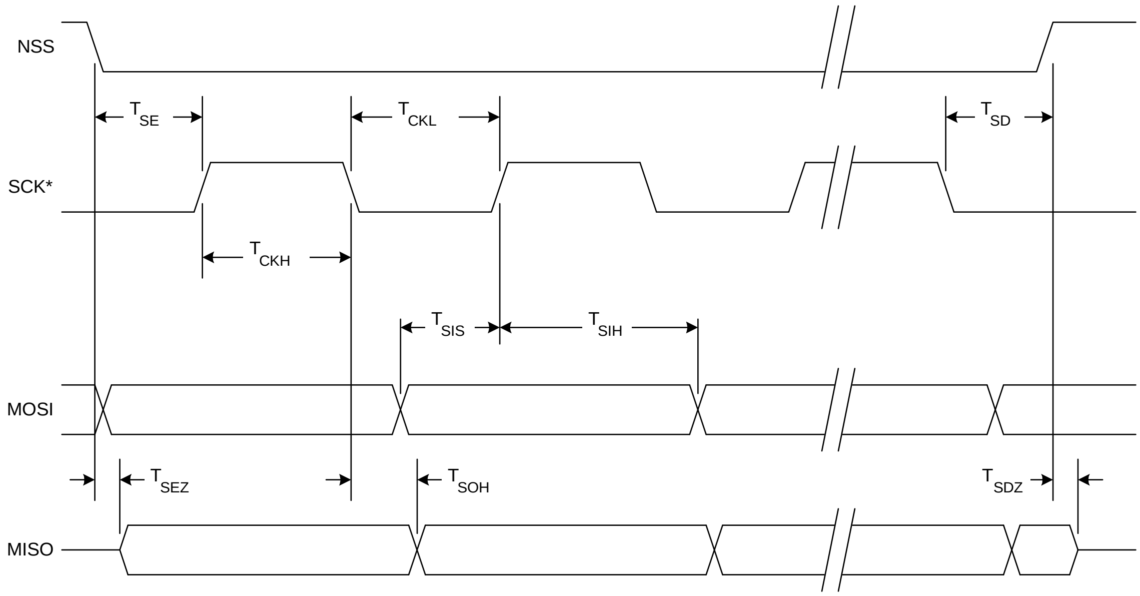
SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 18.4). When a Free Timeout is detected, the interface will respond as if a STOP was detected (an interrupt will be generated, and STO will be set).

## Table 19.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

| | Frequency: 24.5 MHz | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Target Baud Rate (bps)** | **Baud Rate % Error** | **Oscillator Divide Factor** | **Timer Clock Source** | **SCA1–SCA0 (pre-scale select)[1]** | **T1M[1]** | **Timer 1 Reload Value (hex)** |
| SYSCLK from Internal Osc. | 230400 | –0.32% | 106 | SYSCLK | XX[2] | 1 | 0xCB |
| | 115200 | –0.32% | 212 | SYSCLK | XX | 1 | 0x96 |
| | 57600 | 0.15% | 426 | SYSCLK | XX | 1 | 0x2B |
| | 28800 | –0.32% | 848 | SYSCLK/4 | 01 | 0 | 0x96 |
| | 14400 | 0.15% | 1704 | SYSCLK/12 | 00 | 0 | 0xB9 |
| | 9600 | –0.32% | 2544 | SYSCLK/12 | 00 | 0 | 0x96 |
| | 2400 | –0.32% | 10176 | SYSCLK/48 | 10 | 0 | 0x96 |
| | 1200 | 0.15% | 20448 | SYSCLK/48 | 10 | 0 | 0x2B |

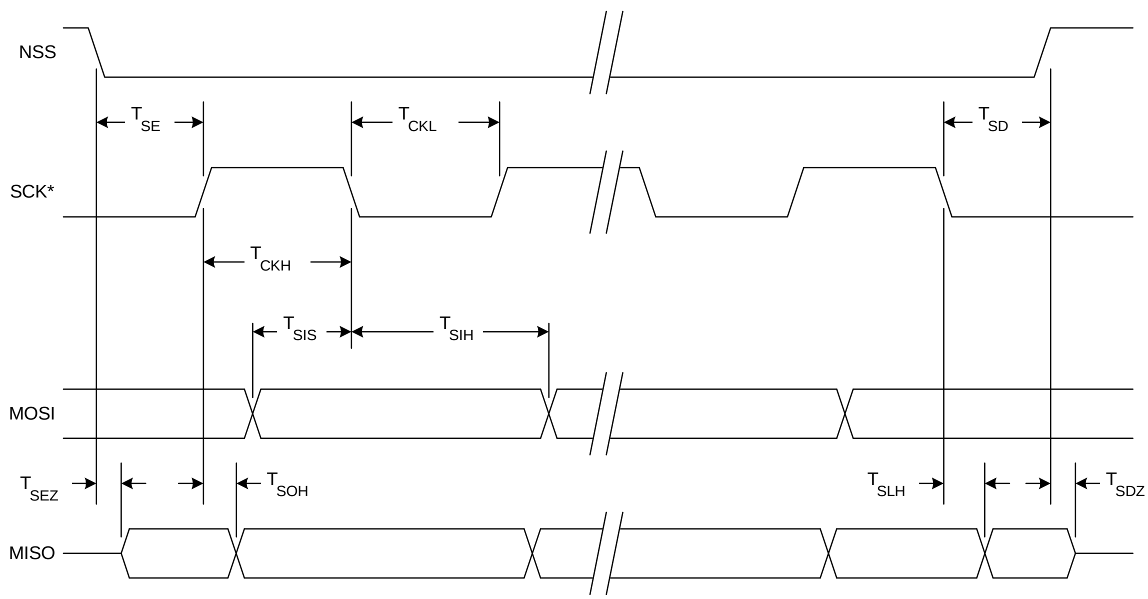
**Notes:**

1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 20.10. SPI Slave Timing (CKPHA = 0)**




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 20.11. SPI Slave Timing (CKPHA = 1)**

### 21.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 21.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

| T2MH | T2XCLK | TMR2H Clock Source |
|---|---|---|
| 0 | 0 | SYSCLK/12 |
| 0 | 1 | External Clock/8 |
| 1 | X | SYSCLK |

| T2ML | T2XCLK | TMR2L Clock Source |
|---|---|---|
| 0 | 0 | SYSCLK/12 |
| 0 | 1 | External Clock/8 |
| 1 | X | SYSCLK |

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

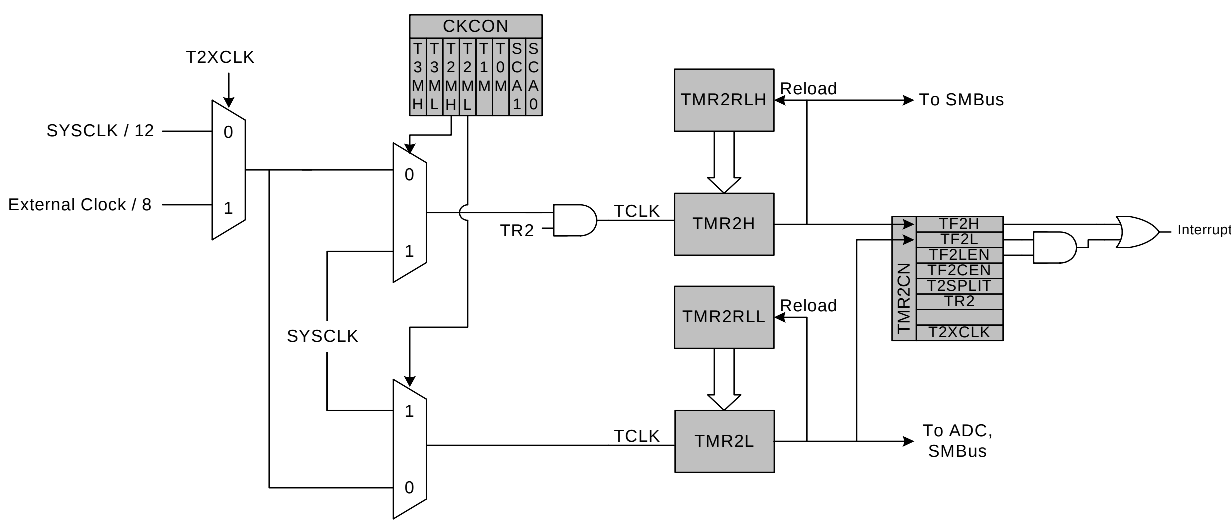



**Figure 21.5. Timer 2 8-Bit Mode Block Diagram**

### 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’.

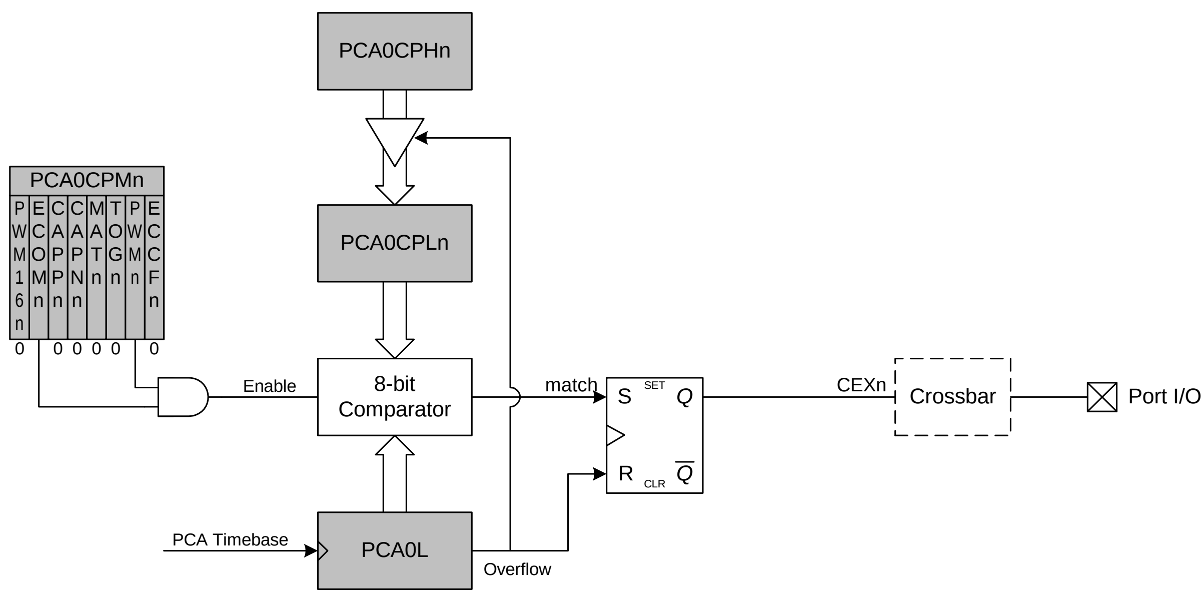



**Figure 22.8. PCA 8-Bit PWM Mode Diagram**