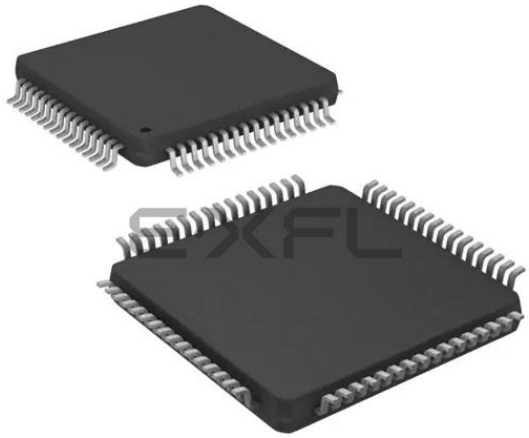

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
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 ~ 3.6V
Data Converters	A/D 8x8b, 13x10b; D/A 2x10b, 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f043r

SFR Definition 5.7. ADC0H: ADC0 Data Word MSB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:

SFR Address: 0xBF
SFR Page: 0

Bits7-0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0: Bits 7-4 are the sign extension of Bit3. Bits 3-0 are the upper 4 bits of the 12-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7-0 are the most-significant bits of the 12-bit ADC0 Data Word.

SFR Definition 5.8. ADC0L: ADC0 Data Word LSB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:

SFR Address: 0xBE
SFR Page: 0

Bits7-0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7-0 are the lower 8 bits of the 12-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7-4 are the lower 4 bits of the 12-bit ADC0 Data Word. Bits3-0 will always read ‘0’.

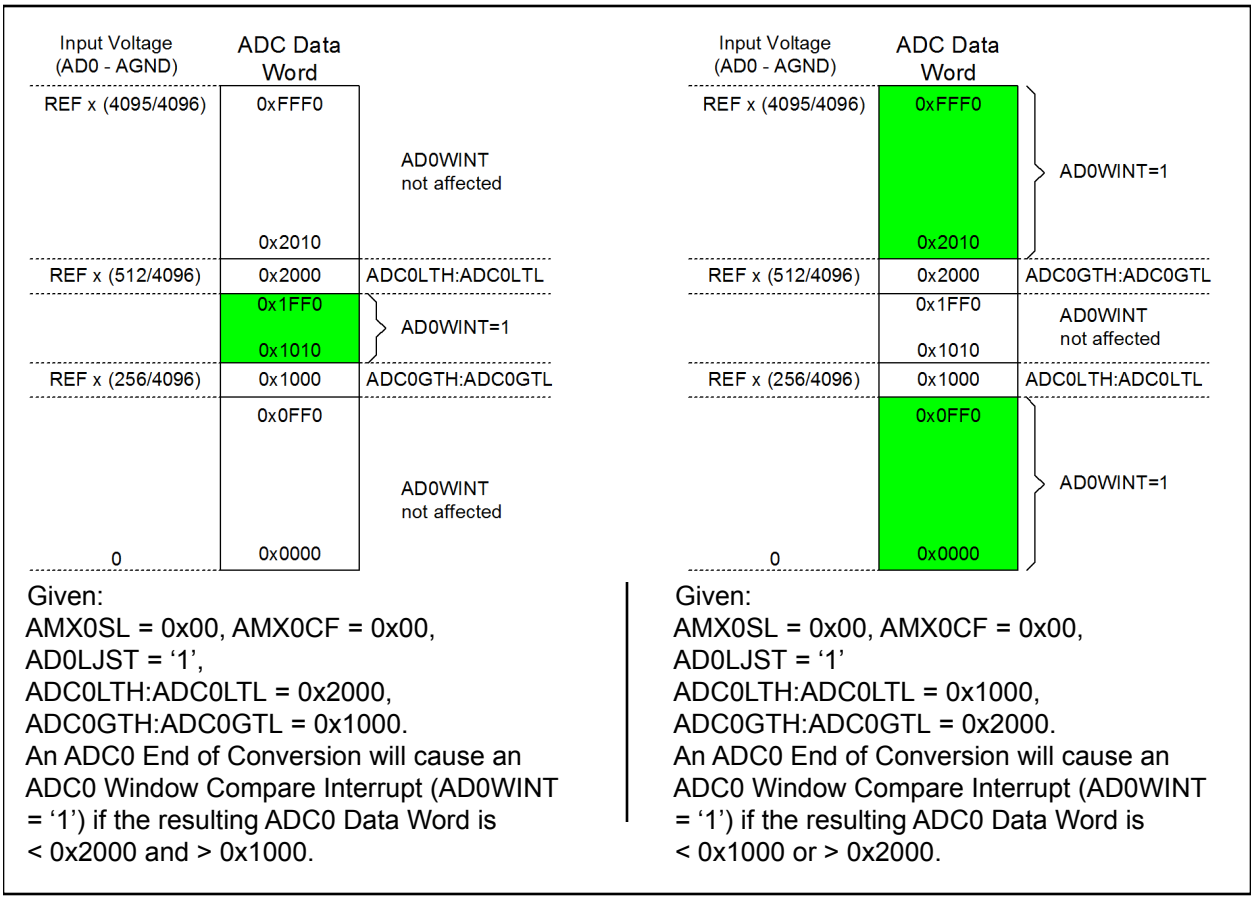

Figure 5.10. 12-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data

6.1.1. Analog Input Configuration

The analog multiplexer routes signals from external analog input pins, Port 3 I/O pins (programmed to be analog inputs), a High Voltage Difference Amplifier, and an on-chip temperature sensor as shown in Figure 6.2.

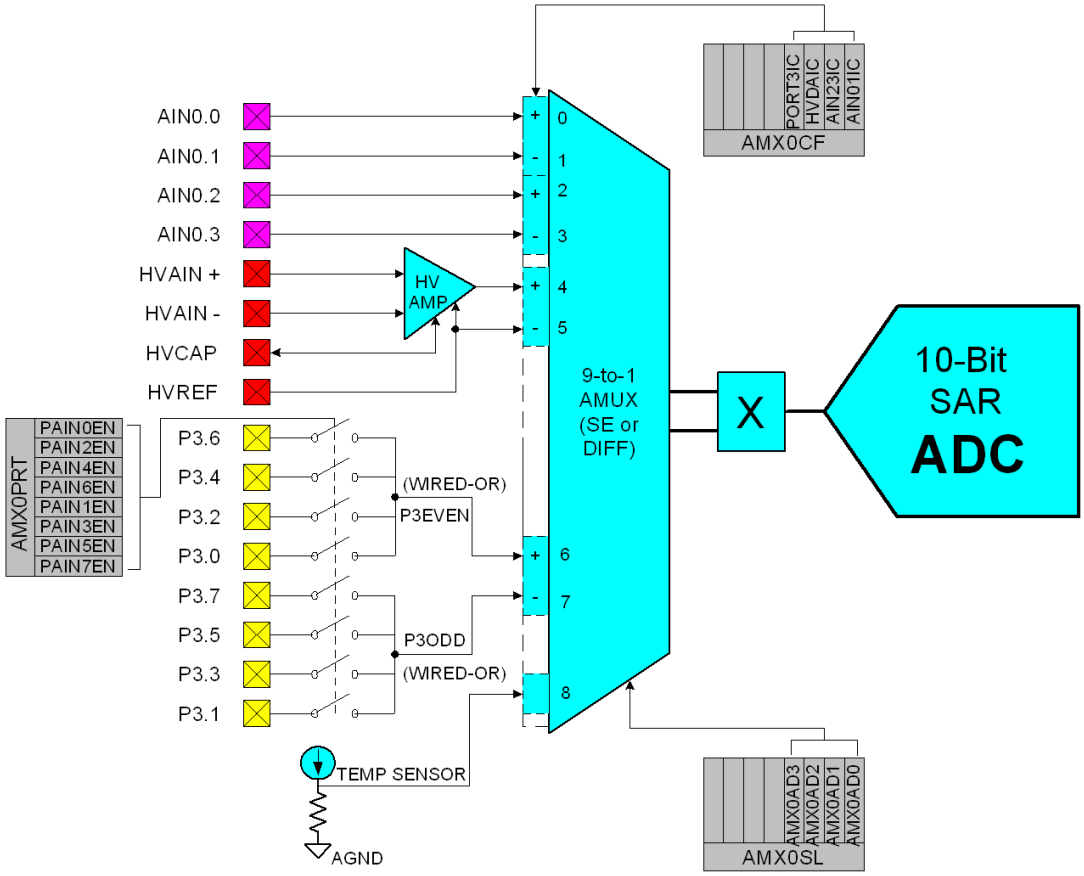

Figure 6.2. Analog Input Diagram

Analog signals may be input from four external analog input pins (AIN0.0 through AIN0.3) as differential or single-ended measurements. Additionally, Port 3 I/O Port Pins may be configured to input analog signals. Port 3 pins configured as analog inputs are selected using the Port Pin Selection register (AMX0PRT). Any number of Port 3 pins may be selected simultaneously as inputs to the AMUX. Even numbered Port 3 pins and odd numbered Port 3 pins are routed to separate AMUX inputs. (**Note:** Even port pins and odd port pins that are simultaneously selected will be shorted together as “wired-OR”.) In this way, differential measurements may be made when using the Port 3 pins (voltage difference between selected even and odd Port 3 pins) as shown in Figure 6.2.

The High-Voltage Difference Amplifier (HVDA) will accept analog input signals and reject up to 60 volts common-mode for differential measurement of up to the reference voltage to the ADC (0 to VREF volts). The output of the HVDA can be selected as an input to the ADC using the AMUX as any other channel is selected for measurement.

7. 8-Bit ADC (ADC2, C8051F040/1/2/3 Only)

The ADC2 subsystem for the C8051F040/1/2/3 consists of an 8-channel, configurable analog multiplexer, a programmable gain amplifier, and a 500 ksps, 8-bit successive-approximation-register ADC with integrated track-and-hold (see block diagram in Figure 7.1). The AMUX2, PGA2, and Data Conversion Modes, are all configurable under software control via the Special Function Registers shown in Figure 7.1. The ADC2 subsystem (8-bit ADC, track-and-hold and PGA) is enabled only when the AD2EN bit in the ADC2 Control register (ADC2CN) is set to logic 1. The ADC2 subsystem is in low power shutdown when this bit is logic 0. The voltage reference used by ADC2 is selected as described in **Section "9. Voltage Reference (C8051F040/2/4/6)" on page 113** for C8051F040/2 devices, or **Section "10. Voltage Reference (C8051F041/3/5/7)" on page 117** for C8051F041/3 devices.

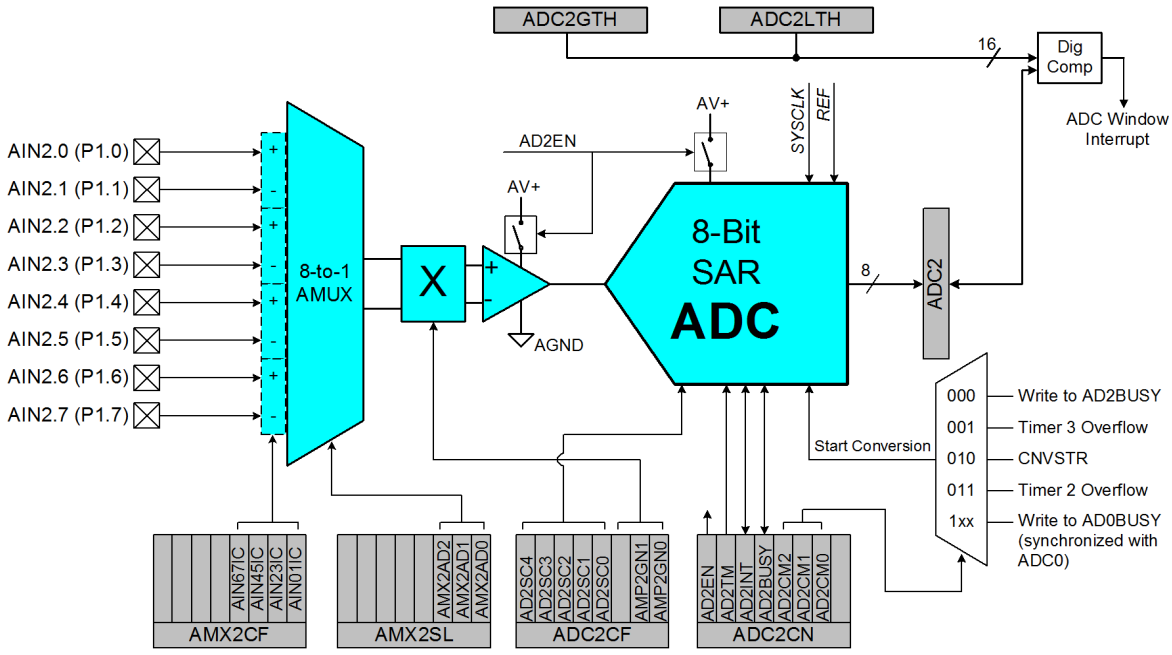

Figure 7.1. ADC2 Functional Block Diagram

7.1. Analog Multiplexer and PGA

Eight ADC2 channels are available for measurement, as selected by the AMX2SL register (see SFR Definition 7.2). The PGA amplifies the ADC2 output signal by an amount determined by the states of the AMP2GN2-0 bits in the ADC2 Configuration register, ADC2CF (SFR Definition 7.1). The PGA can be software-programmed for gains of 0.5, 1, 2, or 4. Gain defaults to 0.5 on reset.

Important Note: AIN2 pins also function as Port 1 I/O pins, and must be configured as analog inputs when used as ADC2 inputs. To configure an AIN2 pin for analog input, set to '0' the corresponding bit in register P1MDIN. Port 1 pins selected as analog inputs are skipped by the Digital I/O Crossbar. See **Section "17.1.5. Configuring Port 1, 2, and 3 Pins as Analog Inputs" on page 207** for more information on configuring the AIN2 pins.

and a RET pops two record bits, also.) The stack record circuitry can also detect an overflow or underflow on the 32-bit shift register, and can notify the debug software even with the MCU running at speed.

12.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFR's). The SFR's provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFR's found in a typical 8051 implementation as well as implementing additional SFR's used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 12.2 lists the SFR's implemented in the CIP-51 System Controller.

The SFR registers are accessed whenever the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFR's with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFR's are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the datasheet, as indicated in Table 12.3, for a detailed description of each register.

12.2.6.1. SFR Paging

The CIP-51 features *SFR paging,* allowing the device to map many SFR's into the 0x80 to 0xFF memory address space. The SFR memory space has 256 *pages*. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFR's. The C8051F04x family of devices utilizes five SFR pages: 0, 1, 2, 3, and F. SFR pages are selected using the Special Function Register Page Selection register, SFRPAGE (see SFR Definition 12.2). The procedure for reading and writing an SFR is as follows:

1. Select the appropriate SFR page number using the SFRPAGE register.
2. Use direct accessing mode to read or write the special function register (MOV instruction).

12.2.6.2. Interrupts and SFR Paging

When an interrupt occurs, the SFR Page Register will automatically switch to the SFR page containing the flag bit that caused the interrupt. The automatic SFR Page switch function conveniently removes the burden of switching SFR pages from the interrupt service routine. Upon execution of the RETI instruction, the SFR page is automatically restored to the SFR Page in use prior to the interrupt. This is accomplished via a three-byte *SFR Page Stack*. The top byte of the stack is SFRPAGE, the current SFR Page. The second byte of the SFR Page Stack is SFRNEXT. The third, or bottom byte of the SFR Page Stack is SFRLAST. On interrupt, the current SFRPAGE value is pushed to the SFRNEXT byte, and the value of SFRNEXT is pushed to SFRLAST. Hardware then loads SFRPAGE with the SFR Page containing the flag bit associated with the interrupt. On a return from interrupt, the SFR Page Stack is popped resulting in the value of SFRNEXT returning to the SFRPAGE register, thereby restoring the SFR page context without software intervention. The value in SFRLAST (0x00 if there is no SFR Page value in the bottom of the stack) of the stack is placed in SFRNEXT register. If desired, the values stored in SFRNEXT and SFRLAST may be modified during an interrupt, enabling the CPU to return to a different SFR Page upon execution of the RETI instruction (on interrupt exit). Modifying registers in the SFR Page Stack does not cause a push or pop of the stack. Only interrupt calls and returns will cause push/pop operations on the SFR Page Stack.

Table 12.4. Interrupt Summary (Continued)

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	SFRPAGE (SFRPGEN = 1)	Enable Flag	Priority Control
Comparator 1	0x005B	11	CP1FIF/CP1RIF (CPT1CN.4/.5)			2	CP1IE (EIE1.5)	PCP1 (EIP1.5)
Comparator 2	0x0063	12	CP2FIF/CP2RIF (CPT2CN.4/.5)			3	CP2IE (EIE1.6)	PCP2 (EIP1.6)
Timer 3	0x0073	14	TF3 (TMR3CN.7)			1	ET3 (EIE2.0)	PT3 (EIP2.0)
ADC0 End of Conversion	0x007B	15	ADC0INT (ADC0CN.5)	Y		0	EADC0 (EIE2.1)	PADC0 (EIP2.1)
Timer 4	0x0083	16	TF4 (TMR4CN.7)			2	ET4 (EIE2.2)	PT4 (EIP2.2)
ADC2 Window Comparator	0x0093	17	AD2WINT (ADC2CN.0)			2	EWADC2 (EIE2.3)	PWADC2 (EIP2.3)
ADC2 End of Conversion	0x008B	18	ADC2INT (ADC1CN.5)			2	EADC1 (EIE2.4)	PADC1 (EIP2.4)
CAN Interrupt	0x009B	19	CAN0CN.7		Y	1	ECAN0 (EIE2.5)	PCAN0 (EIP2.5)
UART1	0x00A3	20	RI1 (SCON1.0) TI1 (SCON1.1)			1	ES1 (EIE2.6)	PS1 (EIP2.6)

12.17. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the external peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. SFR Definition 12.18 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

Although the CIP-51 has Idle and Stop modes built in (as with any standard 8051 architecture), power management of the entire MCU is better accomplished by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and put into low power mode. Digital peripherals, such as timers or serial buses, draw little power whenever they are not in use. Turning off the oscillator saves even more power, but requires a reset to restart the MCU.

12.17.1.Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt or /RST is asserted. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

If enabled, the WDT will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to **Section 13.7** for more information on the use and configuration of the WDT.

Note: Any instruction that sets the IDLE bit should be immediately followed by an instruction that has 2 or more opcode bytes. For example:

```
// in 'C':
PCON |= 0x01;         // set IDLE bit
PCON  = PCON;         // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h        ; set IDLE bit
MOV PCON, PCON        ; ... followed by a 3-cycle dummy instruction
```

If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from IDLE mode when a future interrupt occurs.

12.17.2.Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes. In Stop mode, the CPU and internal oscillators are stopped, effectively shutting down all digital peripherals. Each analog peripheral must be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to sleep for longer than the MCD timeout of 100 µs.

SFR Definition 12.18. PCON: Power Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x87
SFR Page: All Pages

Bits7-3: Reserved.
Bit1: STOP: STOP Mode Select.
Writing a ‘1’ to this bit will place the CIP-51 into STOP mode. This bit will always read ‘0’.
0: No effect.
1: CIP-51 forced into power-down mode. (Turns off internal oscillator).
Bit0: IDLE: IDLE Mode Select.
Writing a ‘1’ to this bit will place the CIP-51 into IDLE mode. This bit will always read ‘0’.
0: No effect.
1: CIP-51 forced into idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, and all peripherals remain active.)

SFR Definition 14.4. OSCXCN: External Oscillator Control

R	R/W	R/W	R/W	R	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8C
SFR Page: F

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

Bits6-4: XOSCMD2-0: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode (External CMOS Clock input on XTAL1 pin).
011: External CMOS Clock Mode with divide by 2 stage (External CMOS Clock input on XTAL1 pin).
10x: RC/C Oscillator Mode with divide by 2 stage.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

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

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

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

CRYSTAL MODE (Circuit from Figure 14.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal frequency.

RC MODE (Circuit from Figure 14.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
$f = 1.23(10^3) / (R \times C)$, where
f = frequency of oscillation in MHz
C = capacitor value in pF
R = Pullup resistor value in kΩ

C MODE (Circuit from Figure 14.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
$f = KF / (C \times V_{DD})$, where
f = frequency of oscillation in MHz
C = capacitor value on XTAL1, XTAL2 pins in pF
V_{DD} = Power Supply on MCU in volts

SFR Definition 16.2. EMI0CF: External Memory Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PRTSEL	EMD2	EMD1	EMD0	EALE1	EALE0	00000011
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA3
SFR Page: 0

Bits7-6: Unused. Read = 00b. Write = don't care.
Bit5: PRTSEL: EMIF Port Select.
0: EMIF active on P0-P3.
1: EMIF active on P4-P7.
Bit4: EMD2: EMIF Multiplex Mode Select.
0: EMIF operates in multiplexed address/data mode.
1: EMIF operates in non-multiplexed mode (separate address and data pins).
Bits3-2: EMD1-0: EMIF Operating Mode Select.
These bits control the operating mode of the External Memory Interface.
00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space.
01: Split Mode without Bank Select: Accesses below the 4k boundary are directed on-chip. Accesses above the 4k boundary are directed off-chip. 8-bit off-chip MOVX operations use the current contents of the Address High port latches to resolve upper address byte. Note that in order to access off-chip space, EMI0CN must be set to a page that is not contained in the on-chip address space.
10: Split Mode with Bank Select: Accesses below the 4k boundary are directed on-chip. Accesses above the 4k boundary are directed off-chip. 8-bit off-chip MOVX operations use the contents of EMI0CN to determine the high-byte of the address.
11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.
Bits1-0: EALE1-0: ALE Pulse-Width Select Bits (only has effect when EMD2 = 1).
00: ALE high and ALE low pulse width = 1 SYSCLK cycle.
01: ALE high and ALE low pulse width = 2 SYSCLK cycles.
10: ALE high and ALE low pulse width = 3 SYSCLK cycles.
11: ALE high and ALE low pulse width = 4 SYSCLK cycles.

16.4.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 16.2. See **Section "16.6.1. Non-multiplexed Mode" on page 196** for more information about Non-multiplexed operation.

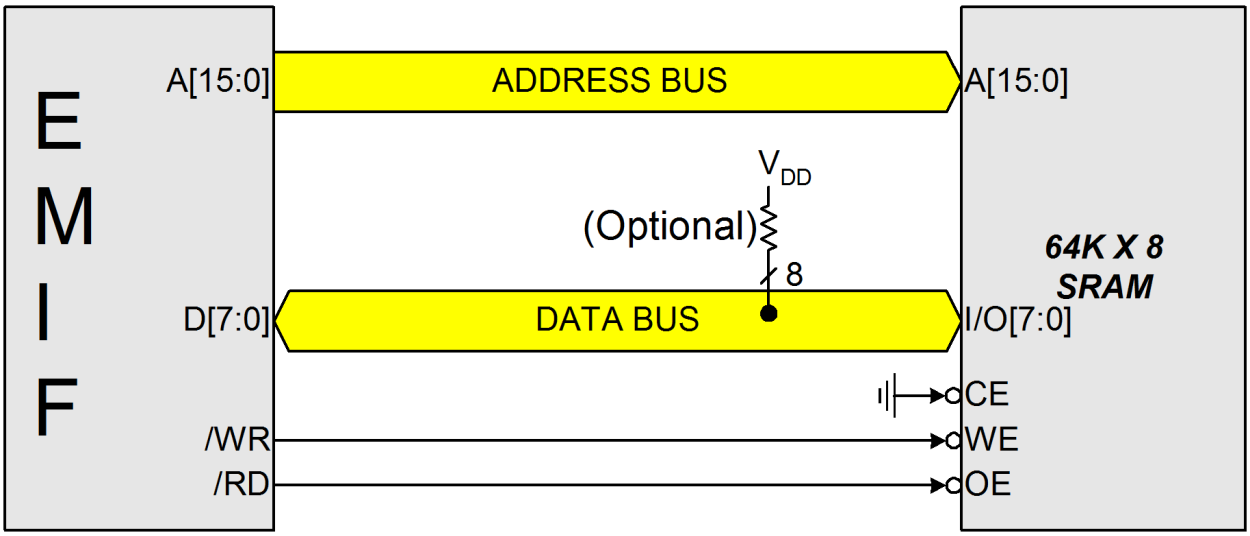

Figure 16.2. Non-multiplexed Configuration Example

16.6.2. Multiplexed Mode

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

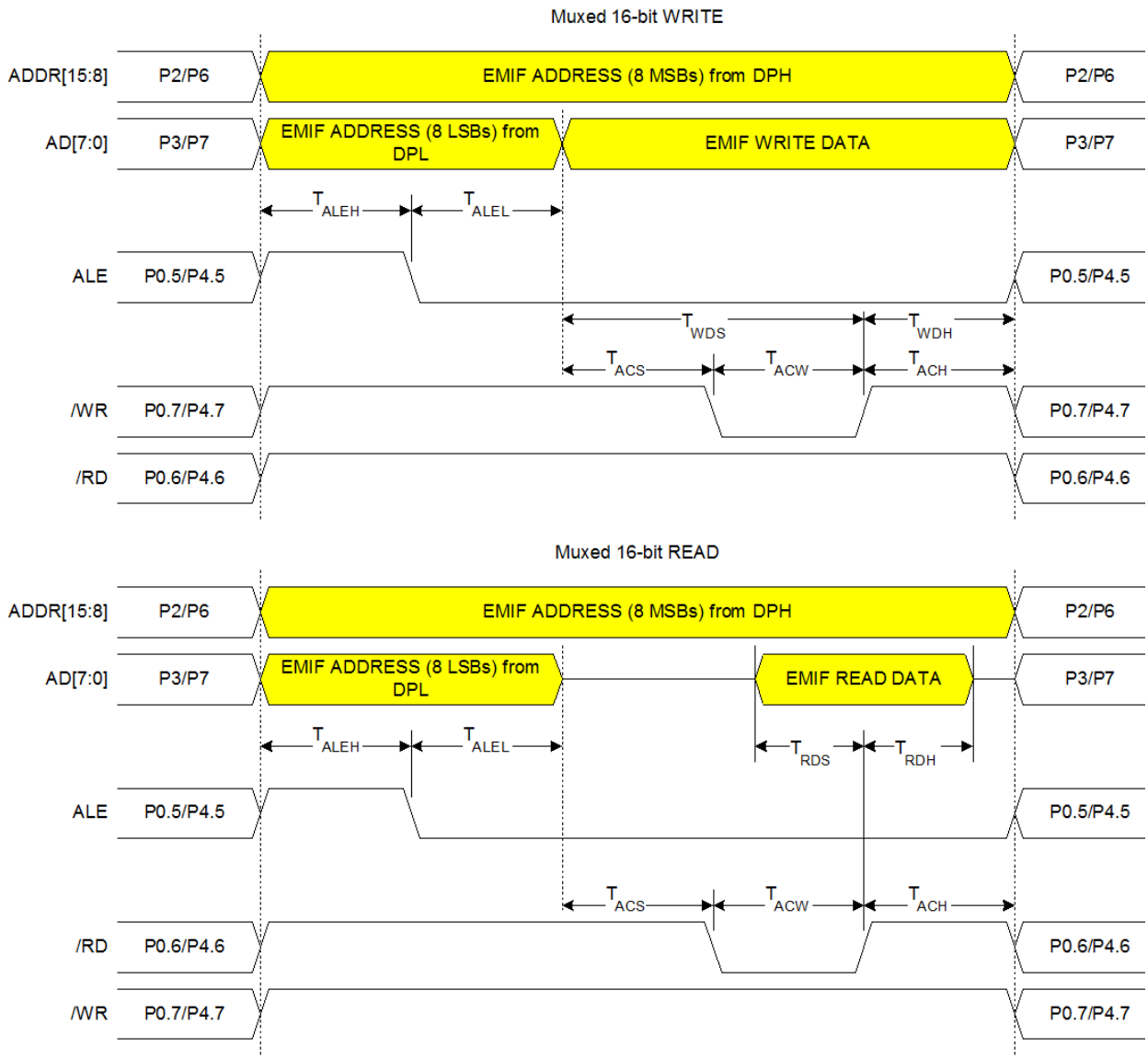

Figure 16.7. Multiplexed 16-bit MOVX Timing

Table 16.1. AC Parameters for External Memory Interface

Parameter	Description	Min	Max	Units
T_{SYSCLK}	System Clock Period	40	—	ns
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

a digital input by setting P3MDOUT.7 to a logic 0, which selects open-drain output mode, and P3.7 to a logic 1, which disables the low-side output driver.

If the Port pin has been assigned to a digital peripheral by the Crossbar and that pin functions as an input (for example RX0, the UART0 receive pin), then the output drivers on that pin are automatically disabled.

17.1.4. Weak Pullups

By default, each Port pin has an internal weak pullup device enabled which provides a resistive connection (about 100 kΩ) between the pin and V_{DD}. The weak pullup devices can be globally disabled by writing a logic 1 to the Weak Pullup Disable bit, (WEAKPUD, XBR2.7). The weak pullup is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pullup device. The weak pullup device can also be explicitly disabled on Ports 1, 2, and 3 pin by configuring the pin as an Analog Input, as described below.

17.1.5. Configuring Port 1, 2, and 3 Pins as Analog Inputs

The pins on Port 1 can serve as analog inputs to the ADC2 analog MUX (C8051F040/1/2/3 only), the pins on Port 2 can serve as analog inputs to the Comparators, and the pins on Port 3 can serve as inputs to ADC0. A Port pin is configured as an Analog Input by writing a logic 0 to the associated bit in the PnMDIN registers. All Port pins default to a Digital Input mode. Configuring a Port pin as an analog input:

1. Disables the digital input path from the pin. This prevents additional power supply current from being drawn when the voltage at the pin is near V_{DD} / 2. A read of the Port Data bit will return a logic 0 regardless of the voltage at the Port pin.
2. Disables the weak pullup device on the pin.
3. Causes the Crossbar to “skip over” the pin when allocating Port pins for digital peripherals, except for P2.0-P2.1.

Note that the output drivers on a pin configured as an Analog Input are not explicitly disabled. Therefore, the associated PnMDOUT bits of pins configured as Analog Inputs should explicitly be set to logic 0 (Open-Drain output mode), and the associated Port Data bits should be set to logic 1 (high-impedance). Also note that it is not required to configure a Port pin as an Analog Input in order to use it as an input to the ADC’s or Comparators; however, it is strongly recommended. See the analog peripheral’s corresponding section in this datasheet for further information.

SFR Definition 18.1. CAN0DATL: CAN Data Access Register Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD8
SFR Page: 1

Bit7-0: CAN0DATL: CAN Data Access Register Low Byte.
The CAN0DAT Registers are used to read/write register values and data to and from the CAN Registers pointed to with the index number in the CAN0ADR Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

SFR Definition 18.2. CAN0ADR: CAN Address Index

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xDA
SFR Page: 1

Bit7-0: CAN0ADR: CAN Address Index Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

Note: When the value of CAN0ADR is 0x08-0x12 and 0x20-0x2A (IF1 and IF2 registers), this register will autoincrement by 1 upon a write to CAN0DATL. See **Section "18.2.6. CAN0ADR Autoincrement Feature" on page 232**.

All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide.

Table 19.1. SMB0STA Status Codes and States

Mode	Status Code	SMBus State	Typical Action
MT/ MR	0x08	START condition transmitted.	Load SMB0DAT with Slave Address + R/W. Clear STA.
	0x10	Repeated START condition transmitted.	Load SMB0DAT with Slave Address + R/W. Clear STA.
Master Transmitter	0x18	Slave Address + W transmitted. ACK received.	Load SMB0DAT with data to be transmitted.
	0x20	Slave Address + W transmitted. NACK received.	Acknowledge poll to retry. Set STO + STA.
	0x28	Data byte transmitted. ACK received.	1) Load SMB0DAT with next byte, OR 2) Set STO, OR 3) Clear STO then set STA for repeated START.
	0x30	Data byte transmitted. NACK received.	1) Retry transfer OR 2) Set STO.
	0x38	Arbitration Lost.	Save current data.
Master Receiver	0x40	Slave Address + R transmitted. ACK received.	If only receiving one byte, clear AA (send NACK after received byte). Wait for received data.
	0x48	Slave Address + R transmitted. NACK received.	Acknowledge poll to retry. Set STO + STA.
	0x50	Data byte received. ACK transmitted.	Read SMB0DAT. Wait for next byte. If next byte is last byte, clear AA.
	0x58	Data byte received. NACK transmitted.	Set STO.

20.5. Serial Clock Timing

As shown in Figure 20.5, four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. Note: SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity.

Note that in master mode, the SPI samples MISO one system clock before the inactive edge of SCK (the edge where MOSI changes state) to provide maximum settling time for the slave device.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 20.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the system clock.

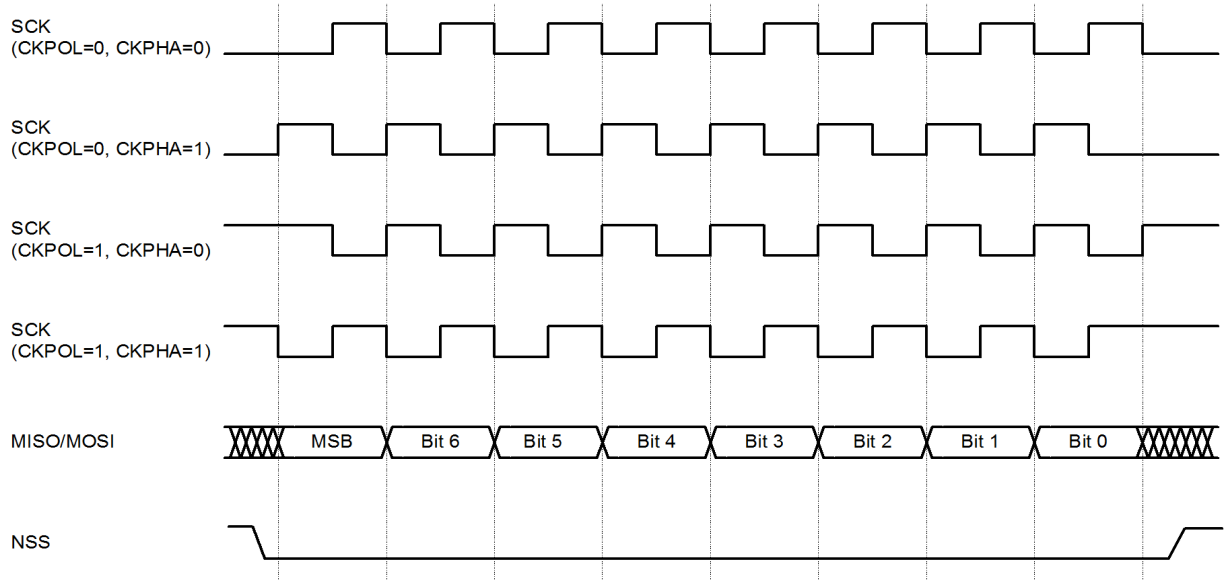

Figure 20.5. Data/Clock Timing Diagram

SFR Definition 23.2. TMOD: Timer Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x89
SFR Page: 0

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 = logic 1.

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5-4: T1M1-T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 = logic 1.

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1-0: T0M1-T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

23.2.2. Capture Mode

In Capture Mode, Timer n will operate as a 16-bit counter/timer with capture facility. When the Timer External Enable bit (found in the TMRnCN register) is set to ‘1’, a high-to-low transition on the TnEX input pin causes the 16-bit value in the associated timer (TMRnH, TMRnL) to be loaded into the capture registers (RCAPnH, RCAPnL). If a capture is triggered in the counter/timer, the Timer External Flag (TMRnCN.6) will be set to ‘1’ and an interrupt will occur if the interrupt is enabled. See **Section “12.3. Interrupt Handler” on page 153** for further information concerning the configuration of interrupt sources.

As the 16-bit timer register increments and overflows TMRnH:TMRnL, the TFn Timer Overflow/Underflow Flag (TMRnCN.7) is set to ‘1’ and an interrupt will occur if the interrupt is enabled. The timer can be configured to count down by setting the Decrement Enable Bit (TMRnCF.0) to ‘1’. This will cause the timer to decrement with every timer clock/count event and underflow when the timer transitions from 0x0000 to 0xFFFF. Just as in overflows, the Overflow/Underflow Flag (TFn) will be set to ‘1’, and an interrupt will occur if enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RLn (TMRnCN.0) and the Timer n Run Control bit TRn (TMRnCN.2) to logic 1. The Timer n respective External Enable EXENn (TMRnCN.3) must also be set to logic 1 to enable captures. If EXENn is cleared, transitions on TnEX will be ignored.

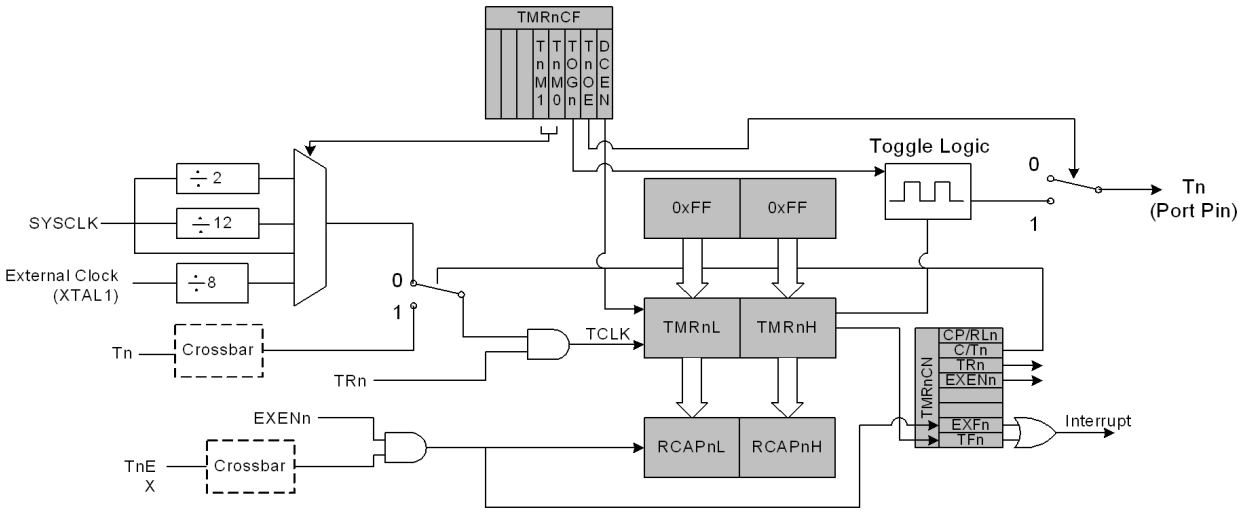

Figure 23.4. Tn Capture Mode Block Diagram

24.2.6. 16-Bit Pulse Width Modulator Mode

Each PCA0 module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA0 clocks for the low time of the PWM signal. When the PCA0 counter matches the module contents, the output on CEXn is asserted high; when the counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA0 CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, CCFn should also be set to logic 1 to enable match interrupts. The duty cycle for 16-Bit PWM Mode is given by Equation 24.3.

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'.

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 24.3. 16-Bit PWM Duty Cycle

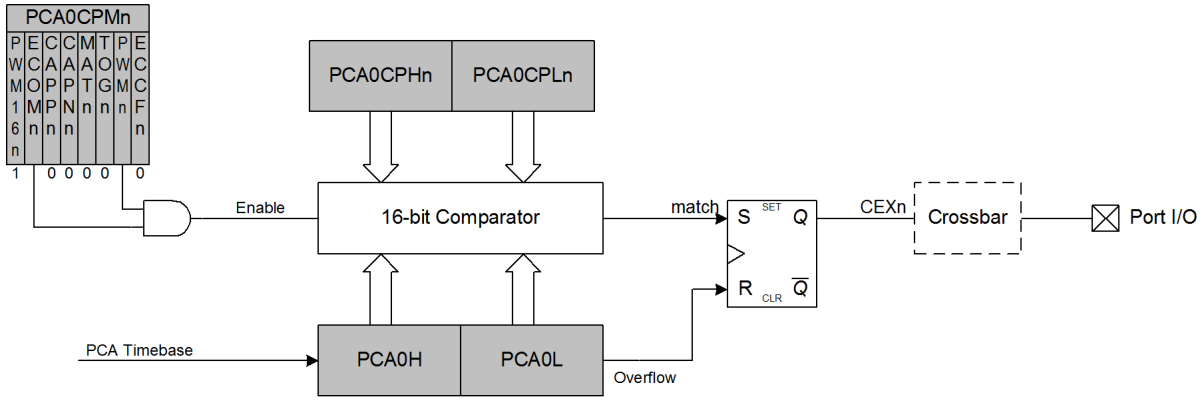

Figure 24.9. PCA 16-Bit PWM Mode