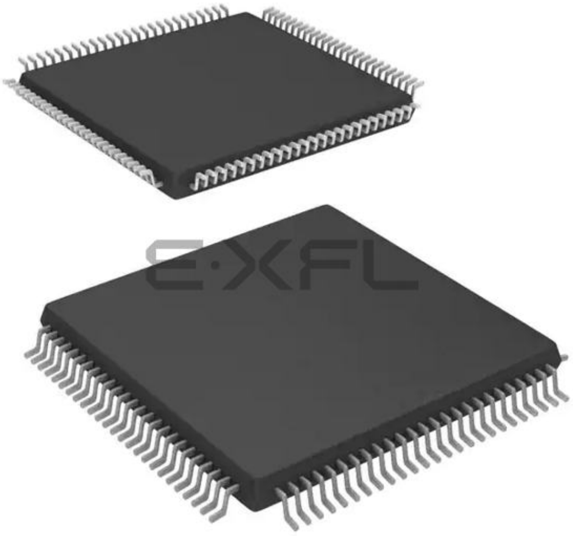
Silicon Labs - C8051F020-GQ Datasheet



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"

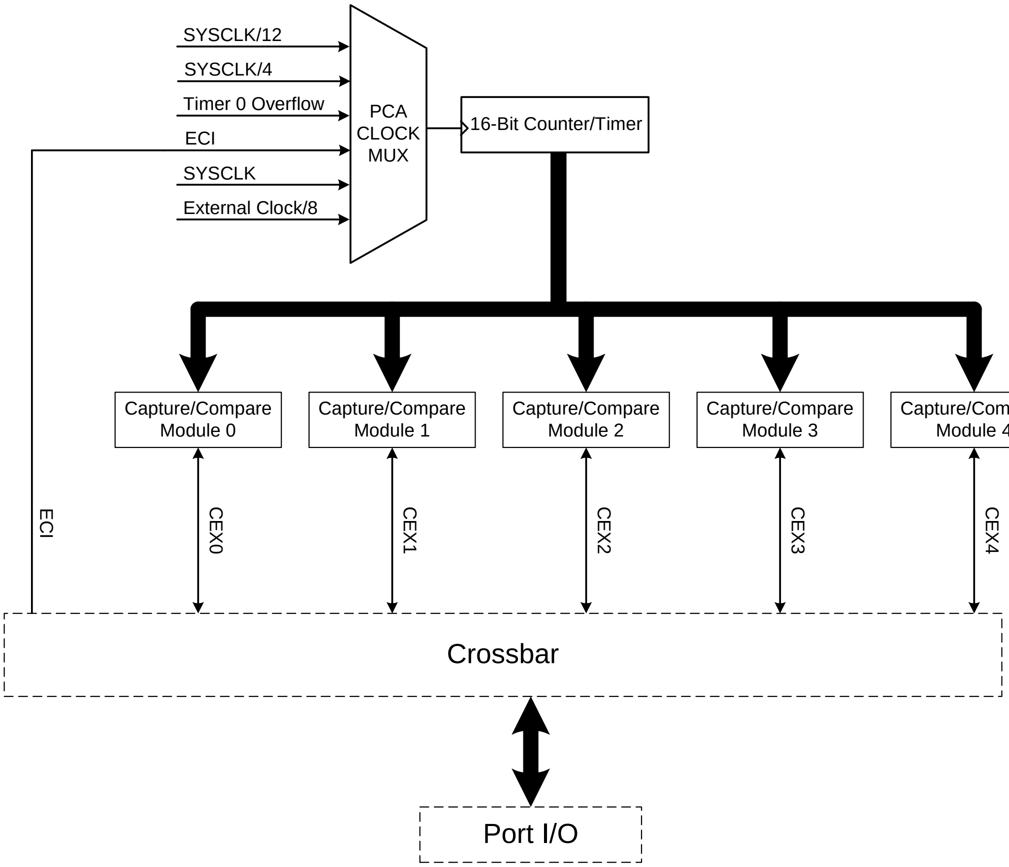
Details	
Product Status	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	64
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x8b, 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f020-gq

1.5. Programmable Counter Array

The C8051F020 MCU family includes an on-board Programmable Counter/Timer Array (PCA) in addition to the five 16-bit general purpose counter/timers. The PCA consists of a dedicated 16-bit counter/timer time base with 5 programmable capture/compare modules. The timebase is clocked from one of six sources: the system clock divided by 12, the system clock divided by 4, Timer 0 overflow, an External Clock Input (ECI pin), the system clock, or the external oscillator source divided by 8.

Each capture/compare module can be configured to operate in one of six modes: Edge-Triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. The PCA Capture/Compare Module I/O and External Clock Input are routed to the MCU Port I/O via the Digital Crossbar.

Figure 1.10. PCA Block Diagram



1.6. Serial Ports

The C8051F020 MCU Family includes two Enhanced Full-Duplex UARTs, SPI Bus, and SMBus/I^{2}C. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little intervention by the CPU. The serial buses do not "share" resources such as timers, interrupts, or Port I/O, so any or all of the serial buses may be used together with any other.

1.9. Comparators and DACs

Each C8051F020/1/2/3 MCU has two 12-bit DACs and two comparators on chip. The MCU data and control interface to each comparator and DAC is via the Special Function Registers. The MCU can place any DAC or comparator in low power shutdown mode.

The comparators have software programmable hysteresis. Each comparator can generate an interrupt on its rising edge, falling edge, or both; these interrupts are capable of waking up the MCU from sleep mode. The comparators' output state can also be polled in software. The comparator outputs can be programmed to appear on the Port I/O pins via the Crossbar.

The DACs are voltage output mode, and include a flexible output scheduling mechanism. This scheduling mechanism allows DAC output updates to be forced by a software write or a Timer 2, 3, or 4 overflow. The DAC voltage reference is supplied via the dedicated VREFD input pin on C8051F020/2 devices or via the internal voltage reference on C8051F021/3 devices. The DACs are especially useful as references for the comparators or offsets for the differential inputs of the ADC.

Figure 1.13. ComparatorandDACDiagram

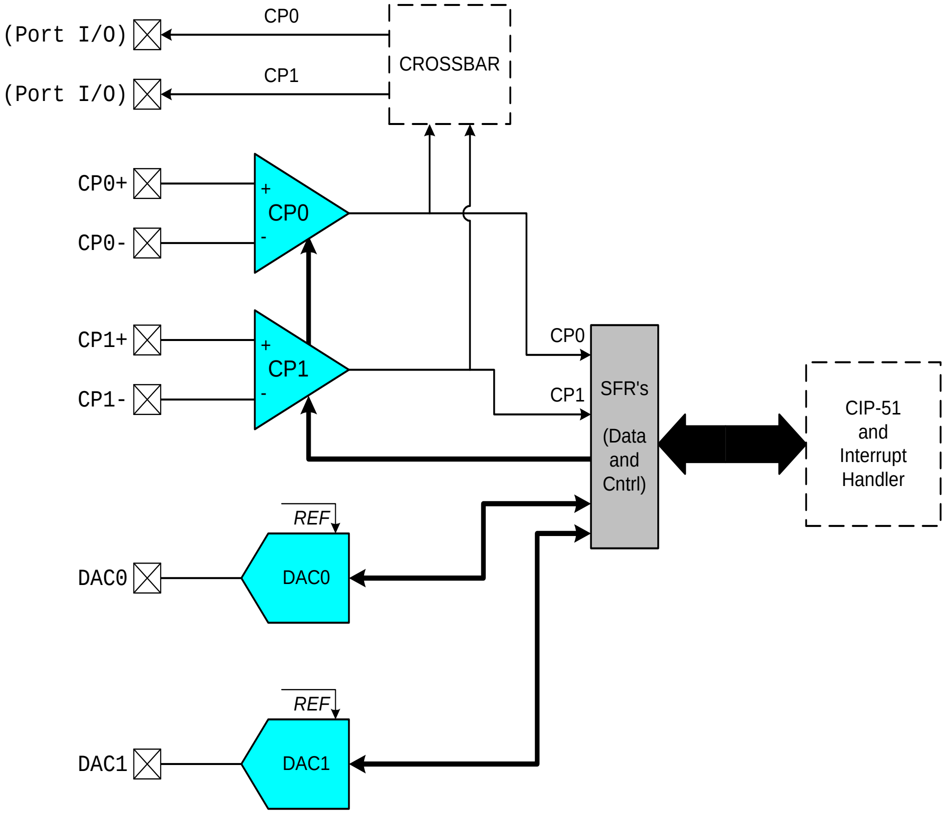


Table 11.1. Comparator Electrical Characteristics

VDD = 3.0 V, AV+ = 3.0 V, -40°C to +85°C unless otherwise specified

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Response Time 1	CP+ - CP- = 100 mV		4		µs
Response Time 2	CP+ - CP- = 10 mV		12		µs
Common-Mode Rejection Ratio			1.5	4	mV/V
Positive Hysteresis 1	CPnHYP1-0 = 00		0	1	mV
Positive Hysteresis 2	CPnHYP1-0 = 01	2	4.5	7	mV
Positive Hysteresis 3	CPnHYP1-0 = 10	4	9	13	mV
Positive Hysteresis 4	CPnHYP1-0 = 11	10	17	25	mV
Negative Hysteresis 1	CPnHYN1-0 = 00		0	1	mV
Negative Hysteresis 2	CPnHYN1-0 = 01	2	4.5	7	mV
Negative Hysteresis 3	CPnHYN1-0 = 10	4	9	13	mV
Negative Hysteresis 4	CPnHYN1-0 = 11	10	17	25	mV
Inverting or Non-Inverting Input Voltage Range		-0.25		(AV+) + 0.25	V
Input Capacitance			7		pF
Input Bias Current		-5	0.001	+5	nA
Input Offset Voltage		-10		+10	mV
POWER SUPPLY					
Power-up Time	CPnEN from 0 to 1		20		µs
Power Supply Rejection			0.1	1	mV/V
Supply Current	Operating Mode (each comparator) at DC		1.5	10	µA

Notes on Registers, Operands and Addressing Modes:
Rn - Register R0-R7 of the currently selected register bank. @Ri - Data RAM location addressed indirectly through R0 or R1. **rel** - 8-bit, signed (two's complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps. **direct** - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF). **#data** - 8-bit constant **#data16** - 16-bit constant **bit** - Direct-accessed bit in Data RAM or SFR **addr11** - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-byte page of program memory as the first byte of the following instruction. **addr16** - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space. There is one unused opcode (0xA5) that performs the same function as NOP. All mnemonics copyrighted © Intel Corporation 1980.

12.2.2. Data Memory

The CIP-51 implements 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 12.2 illustrates the data memory organization of the CIP-51.

12.2.3. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in Figure 12.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

12.2.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

12.2.5. Stack

A programmer's stack can be located anywhere in the 256 byte data memory. The stack area is designated using the Stack Pointer (SP, address 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07; therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

The MCUs also have built-in hardware for a stack record. The stack record is a 32-bit shift register, where each PUSH or increment SP pushes one record bit onto the register, and each CALL pushes two record bits onto the register. (A POP or decrement SP pops one record bit, 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.

Figure 12.12. EIE2: Extended Interrupt Enable 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EXVLD	ES1	EX7	EX6	EADC1	ET4	EADC0	ET3	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xE7

Bit7: EXVLD: Enable External Clock Source Valid (XTLVLD) Interrupt.
This bit sets the masking of the XTLVLD interrupt.
0: Disable XTLVLD interrupt.
1: Enable interrupt requests generated by the XTLVLD flag (OSCXCN.7)

Bit6: ES1: Enable UART1 Interrupt.
This bit sets the masking of the UART1 interrupt.
0: Disable UART1 interrupt.
1: Enable UART1 interrupt.

Bit5: EX7: Enable External Interrupt 7.
This bit sets the masking of External Interrupt 7.
0: Disable External Interrupt 7.
1: Enable interrupt requests generated by the External Interrupt 7 input pin.

Bit4: EX6: Enable External Interrupt 6.
This bit sets the masking of External Interrupt 6.
0: Disable External Interrupt 6.
1: Enable interrupt requests generated by the External Interrupt 6 input pin.

Bit3: EADC1: Enable ADC1 End Of Conversion Interrupt.
This bit sets the masking of the ADC1 End of Conversion interrupt.
0: Disable ADC1 End of Conversion interrupt.
1: Enable interrupt requests generated by the ADC1 End of Conversion Interrupt.

Bit2: ET4: Enable Timer 4 Interrupt
This bit sets the masking of the Timer 4 interrupt.
0: Disable Timer 4 interrupt.
1: Enable interrupt requests generated by the TF4 flag (T4CON.7).

Bit1: EADC0: Enable ADC0 End of Conversion Interrupt.
This bit sets the masking of the ADC0 End of Conversion Interrupt.
0: Disable ADC0 Conversion Interrupt.
1: Enable interrupt requests generated by the ADC0 Conversion Interrupt.

Bit0: ET3: Enable Timer 3 Interrupt.
This bit sets the masking of the Timer 3 interrupt.
0: Disable all Timer 3 interrupts.
1: Enable interrupt requests generated by the TF3 flag (TMR3CN.7).

14.1. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be as shown in Figure 14.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in Figure 14.3 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

The Crystal Oscillator Valid Flag (XTLVLD in register OSCXCN) is set to logic 1 by hardware when the external crystal oscillator is running and stable. The XTLVLD detection circuit requires a startup time of at least 1 ms between enabling the oscillator and checking the XTLVLD bit. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Enable the external oscillator.
Step 2. Wait at least 1 ms.
Step 3. Poll for XTLVLD => ‘1’.
Step 4. Switch the system clock to the external oscillator.

Important Note: 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, as should the loading capacitors on the crystal pins. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

14.2. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be as shown in Figure 14.1, Option 2. The capacitor must be no greater than 100 pF; however for small capacitors (less than ~20 pF), the total capacitance may be dominated by PWB parasitic capacitance. 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:

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

$XFCN \geq \log_2 (f / 25 \text{ kHz})$
$XFCN \geq \log_2 (100 \text{ kHz} / 25 \text{ kHz}) = \log_2 (4)$
$XFCN \geq 2$, or code 010b

14.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be as shown in Figure 14.1, Option 3. The capacitor must be no greater than 100 pF; however for small capacitors (less than ~20 pF), the total capacitance may be dominated by PWB parasitic capacitance. 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 VDD = 3.0 V and C = 50 pF:

$f = KF / (C * VDD) = KF / (50 * 3)$
$f = KF / 150$

If a frequency of roughly 90 kHz is desired, select the K Factor from the table in Figure 14.3 as KF = 13:

$f = 13 / 150 = 0.087$ MHz, or 87 kHz

Therefore, the XFCN value to use in this example is 011b.

Figure 15.4. PSCTL: Program Store Read/Write Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	SFLE	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8F

Bits7-3: UNUSED. Read = 00000b, Write = don't care.

Bit2: SFLE: Scratchpad FLASH Memory Access Enable.
When this bit is set, FLASH reads and writes from user software are directed to the 128-byte Scratchpad FLASH sector. When SFLE is set to logic 1, FLASH accesses out of the address range 0x00-0x7F should not be attempted. Reads/Writes out of this range will yield unpredictable results.
0: FLASH access from user software directed to the 64k byte Program/Data FLASH sector.
1: FLASH access from user software directed to the 128 byte Scratchpad sector.

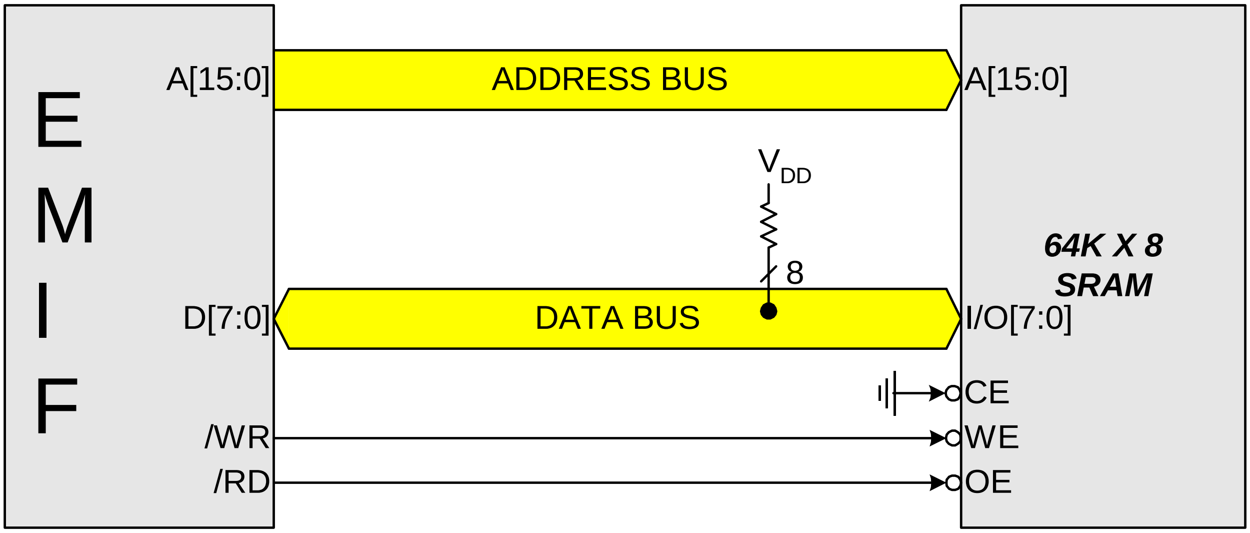
Bit1: PSEE: Program Store Erase Enable.
Setting this bit allows an entire page of the FLASH program memory to be erased provided the PSWE bit is also set. After setting this bit, a write to FLASH memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter.
0: FLASH program memory erasure disabled.
1: FLASH program memory erasure enabled.

Bit0: PSWE: Program Store Write Enable.
Setting this bit allows writing a byte of data to the FLASH program memory using the MOVX instruction. The location must be erased before writing data.
0: Write to FLASH program memory disabled.
1: Write to FLASH program memory enabled.

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 16.4. See **Section “16.6.1. Non-multiplexed Mode” on page 153** for more information about Non-multiplexed operation.

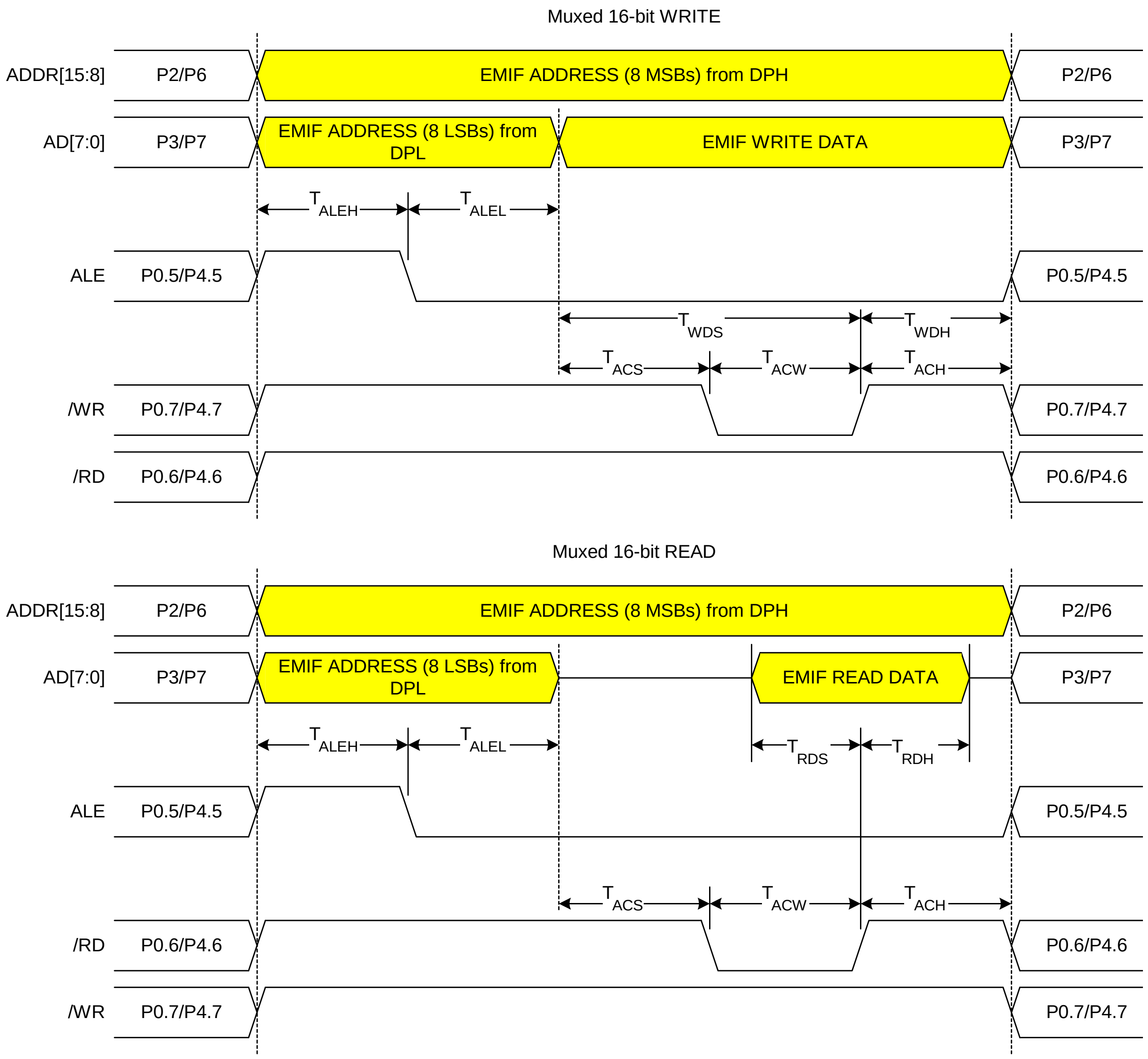
Figure 16.4. Non-multiplexed Configuration Example



16.6.2. Multiplexed Mode

16.6.2.1. 16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

Figure 16.10. Multiplexed 16-bit MOVX Timing



17.1.7. External Memory Interface Pin Assignments

If the External Memory Interface (EMIF) is enabled on the Low ports (Ports 0 through 3), EMIFLE (XBR2.1) should be set to a logic 1 so that the Crossbar will not assign peripherals to P0.7 (/WR), P0.6 (/RD), and if the External Memory Interface is in Multiplexed mode, P0.5 (ALE). Figure 17.4 shows an example Crossbar Decode Table with EMIFLE=1 and the EMIF in Multiplexed mode. Figure 17.5 shows an example Crossbar Decode Table with EMIFLE=1 and the EMIF in Non-multiplexed mode.

If the External Memory Interface is enabled on the Low ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Crossbar registers or the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus. See **Section "16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM" on page 145** for more information about the External Memory Interface.

Figure 17.4. Priority Crossbar Decode Table
EMIFLE = 1; EMIF in Multiplexed Mode; P1MDIN = 0xFF)

	P0								P1								P2								P3								Crossbar Register Bits
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
TX0	●																																UART0EN: XBR0.2
RX0		●																															
SCK	●		●																														SPI0EN: XBR0.1
MISO		●		●																													
MOSI			●		●																												
NSS				●					●																								
SDA	●		●		●					●																							SMB0EN: XBR0.0
SCL		●		●					●		●																						
TX1	●		●		●					●		●																					UART1EN: XBR2.2
RX1		●		●					●		●	●	●																				
CEX0	●		●		●					●		●	●	●																			PCA0ME: XBR0.[5:3]
CEX1		●		●					●		●	●	●	●	●																		
CEX2			●		●					●		●	●	●	●	●																	
CEX3				●					●		●	●	●	●	●	●	●																
CEX4					●					●		●	●	●	●	●	●	●															
ECI	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●														ECI0E: XBR0.6
CP0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●													CP0E: XBR0.7
CP1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●												CP1E: XBR1.0
T0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●											T0E: XBR1.1
/INT0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										INT0E: XBR1.2
T1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									T1E: XBR1.3
/INT1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								INT1E: XBR1.4
T2	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							T2E: XBR1.5
T2EX	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						T2EXE: XBR1.6
T4	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					T4E: XBR2.3
T4EX	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				T4EXE: XBR2.4
/SYSCLK	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			SYSCKE: XBR1.7
CNVSTR	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		CNVSTE: XBR2.0
						ALE	/RD	/WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A1	AIN1.3/A1	AIN1.4/A1	AIN1.5/A1	AIN1.6/A1	AIN1.7/A1	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN1 Inputs/Non-muxed Addr H								Muxed Addr H/Non-muxed Addr L								Muxed Data/Non-muxed Data								

Figure 17.8. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bit7: SYSCKE: /SYSCLK Output Enable Bit.
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK routed to Port pin.

Bit6: T2EXE: T2EX Input Enable Bit.
0: T2EX unavailable at Port pin.
1: T2EX routed to Port pin.

Bit5: T2E: T2 Input Enable Bit.
0: T2 unavailable at Port pin.
1: T2 routed to Port pin.

Bit4: INT1E: /INT1 Input Enable Bit.
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port pin.

Bit3: T1E: T1 Input Enable Bit.
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.

Bit2: INT0E: /INT0 Input Enable Bit.
0: /INT0 unavailable at Port pin.
1: /INT1 routed to Port pin.

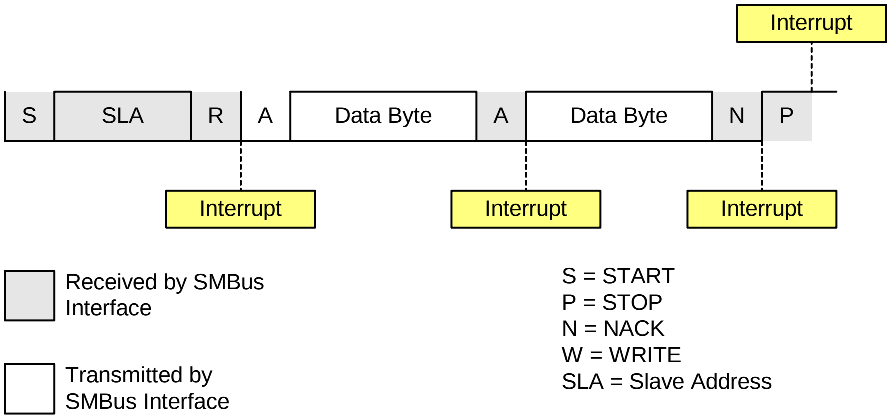
Bit1: T0E: T0 Input Enable Bit.
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.

Bit0: CP1E: CP1 Output Enable Bit.
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

18.3.3. Slave Transmitter Mode

Serial data is transmitted on SDA while the serial clock is received on SCL. The SMBus0 interface receives a START followed by data byte containing the slave address and direction bit. If the received slave address matches the address held in register SMB0ADR, the SMBus0 interface generates an ACK. SMBus0 will also ACK if the general call address (0x00) is received and the General Call Address Enable bit (SMB0ADR.0) is set to logic 1. In this case the data direction bit (R/W) will be logic 1 to indicate a "READ" operation. The SMBus0 interface receives the clock on SCL and transmits one or more bytes of serial data, waiting for an ACK from the master after each byte. SMBus0 exits slave mode after receiving a STOP condition from the master.

Figure 18.6. Typical Slave Transmitter Sequence



18.3.4. Slave Receiver Mode

Serial data is received on SDA while the serial clock is received on SCL. The SMBus0 interface receives a START followed by data byte containing the slave address and direction bit. If the received slave address matches the address held in register SMB0ADR, the interface generates an ACK. SMBus0 will also ACK if the general call address (0x00) is received and the General Call Address Enable bit (SMB0ADR.0) is set to logic 1. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. The SMBus0 interface receives one or more bytes of serial data; after each byte is received, the interface transmits an ACK or NACK depending on the state of the AA bit in SMB0CN. SMBus0 exits Slave Receiver Mode after receiving a STOP condition from the master.

Figure 18.7. Typical Slave Receiver Sequence

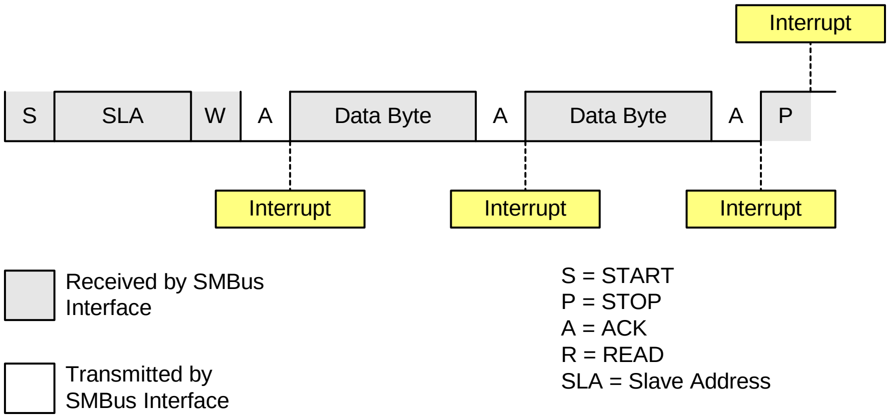


Figure 20.9. SBUF0: UART0 Data Buffer Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x99

Bits7-0: SBUF0.[7:0]: UART0 Buffer Bits 7-0 (MSB-LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 is what initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Figure 20.10. SADDR0: UART0 Slave Address Register

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:
								0xA9

Bits7-0: SADDR0.[7:0]: UART0 Slave Address
The contents of this register are used to define the UART0 slave address. Register SADEN0 is a bit mask to determine which bits of SADDR0 are checked against a received address: corresponding bits set to logic 1 in SADEN0 are checked; corresponding bits set to logic 0 are "don't cares".

Figure 20.11. SADEN0: UART0 Slave Address Enable Register

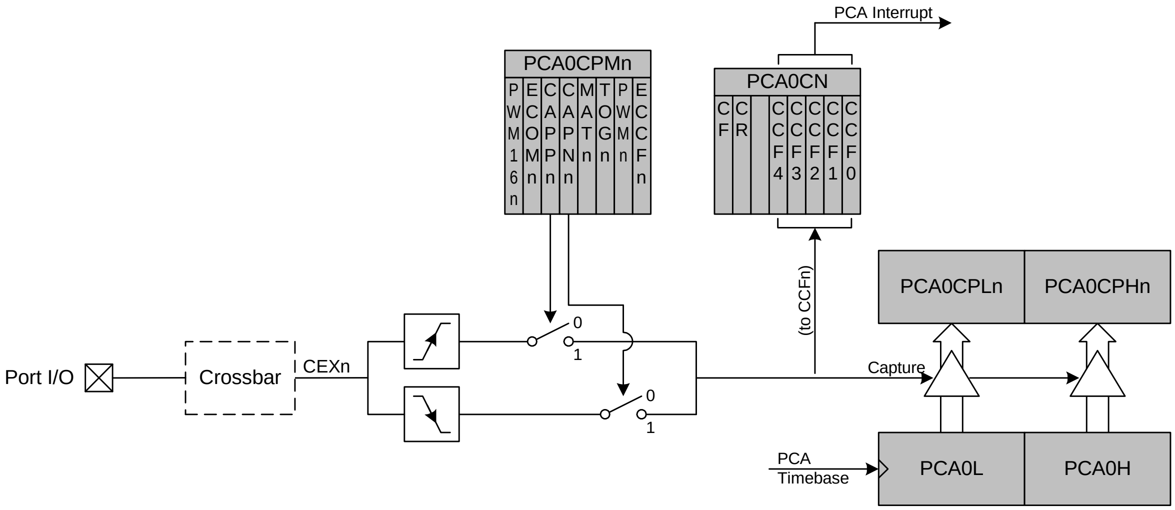
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xB9

Bits7-0: SADEN0.[7:0]: UART0 Slave Address Enable
Bits in this register enable corresponding bits in register SADDR0 to determine the UART0 slave address.
0: Corresponding bit in SADDR0 is a "don't care".
1: Corresponding bit in SADDR0 is checked against a received address.

23.2.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes PCA0 to capture the value of the PCA0 counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software.

Figure 23.4. PCA Capture Mode Diagram



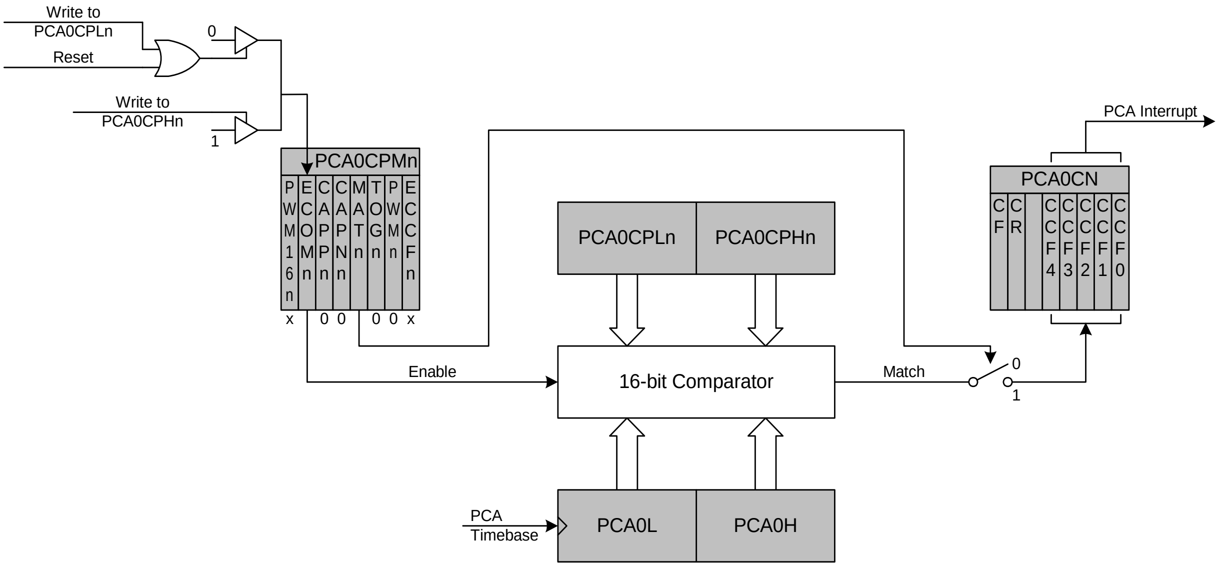
Note: The CEXn input signal must remain high or low for at least 2 system clock cycles in order to be valid.

23.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA0 counter/timer is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to ‘0’; writing to PCA0CPHn sets ECOMn to ‘1’.

Figure 23.5. PCA Software Timer Mode Diagram



23.2.6. 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 23.3.

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’

Equation 23.3. 16-Bit PWM Duty Cycle

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

Using Equation 23.3, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to ‘0’.

Figure 23.9. PCA 16-Bit PWM Mode

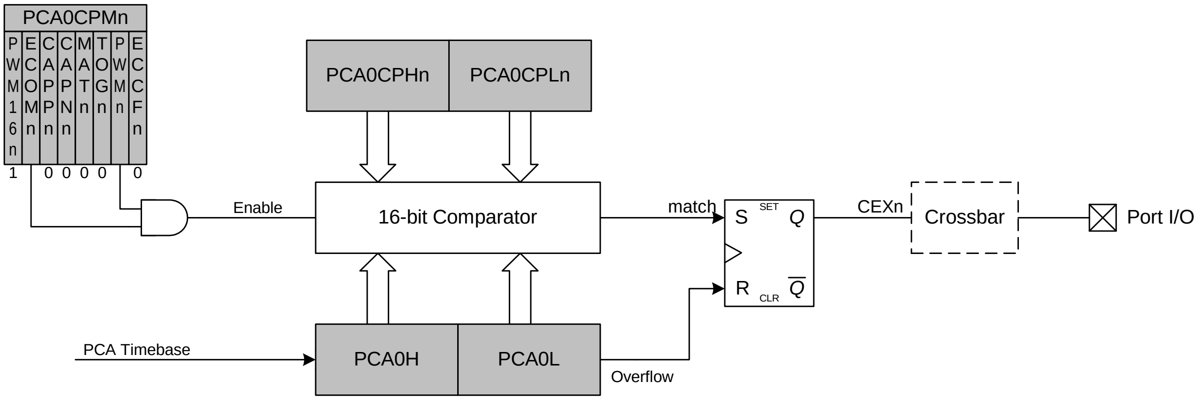


Figure 23.13. PCA0L: PCA0 Counter/Timer Low Byte

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:
								0xE9

Bits 7-0: PCA0L: PCA0 Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA0 Counter/Timer.

Figure 23.14. PCA0H: PCA0 Counter/Timer High Byte

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:
								0xF9

Bits 7-0: PCA0H: PCA0 Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA0 Counter/Timer.

24.2. Flash Programming Commands

The Flash memory can be programmed directly over the JTAG interface using the Flash Control, Flash Data, Flash Address, and Flash Scale registers. These Indirect Data Registers are accessed via the JTAG Instruction Register. Read and write operations on indirect data registers are performed by first setting the appropriate DR address in the IR register. Each read or write is then initiated by writing the appropriate Indirect Operation Code (IndOpCode) to the selected data register. Incoming commands to this register have the following format:

19:18	17:0
IndOpCode	WriteData

IndOpCode: These bit set the operation to perform according to the following table:

IndOpCode	Operation
0x	Poll
10	Read
11	Write

The Poll operation is used to check the Busy bit as described below. Although a Capture-DR is performed, no Update-DR is allowed for the Poll operation. Since updates are disabled, polling can be accomplished by shifting in/ out a single bit.

The Read operation initiates a read from the register addressed by the DRAddress. Reads can be initiated by shifting only 2 bits into the indirect register. After the read operation is initiated, polling of the Busy bit must be performed to determine when the operation is complete.

The write operation initiates a write of WriteData to the register addressed by DRAddress. Registers of any width up to 18 bits can be written. If the register to be written contains fewer than 18 bits, the data in WriteData should be left-justified, i.e. its MSB should occupy bit 17 above. This allows shorter registers to be written in fewer JTAG clock cycles. For example, an 8-bit register could be written by shifting only 10 bits. After a Write is initiated, the Busy bit should be polled to determine when the next operation can be initiated. The contents of the Instruction Register should not be altered while either a read or write operation is busy.

Outgoing data from the indirect Data Register has the following format:

19	18:1	0
0	ReadData	Busy

The Busy bit indicates that the current operation is not complete. It goes high when an operation is initiated and returns low when complete. Read and Write commands are ignored while Busy is high. In fact, if polling for Busy to be low will be followed by another read or write operation, JTAG writes of the next operation can be made while checking for Busy to be low. They will be ignored until Busy is read low, at which time the new operation will initiate. This bit is placed ate bit 0 to allow polling by single-bit shifts. When waiting for a Read to complete and Busy is 0, the following 18 bits can be shifted out to obtain the resulting data. ReadData is always right-justified. This allows registers shorter than 18 bits to be read using a reduced number of shifts. For example, the results from a byte-read requires 9 bit shifts (Busy + 8 bits).