



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	29
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f314-gqr

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

NOTES:



15.UART0
Table 15.1. Timer Settings for Standard Baud Rates
Using the Internal Oscillator
Table 15.2. Timer Settings for Standard Baud Rates
Using an External 25 MHz Oscillator 170
Table 15.3. Timer Settings for Standard Baud Rates
Using an External 22.1184 MHz Oscillator
Table 15.4. Timer Settings for Standard Baud Rates
Using an External 18.432 MHz Oscillator
Table 15.5. Timer Settings for Standard Baud Rates
Using an External 11.0592 MHz Oscillator 172
Table 15.6. Timer Settings for Standard Baud Rates
Using an External 3.6864 MHz Oscillator 172
16. Enhanced Serial Peripheral Interface (SPI0)
Table 16.1. SPI Slave Timing Parameters
17. Timers
18. Programmable Counter Array
Table 18.1. PCA Timebase Input Options 204
Table 18.2. PCA0CPM Register Settings for PCA Capture/Compare Modules 205
Table 18.3. Watchdog Timer Timeout Intervals 214
19. Revision Specific Behavior
20. C2 Interface



1.4. Programmable Digital I/O and Crossbar

C8051F310/2/4 devices include 29 I/O pins (three byte-wide Ports and one 5-bit-wide Port); C8051F311/3/5 devices include 25 I/O pins (three byte-wide Ports and one 1-bit-wide Port); C8051F316/7 devices include 21 I/O pins (one byte-wide Port, two 6-bit-wide Ports and one 1-bit-wide Port). The C8051F31x Ports behave like typical 8051 Ports with a few enhancements. Each Port pin may be configured as an analog input or a digital I/O pin. Pins selected as digital I/Os may additionally be configured for push-pull or open-drain output. The "weak pullups" that are fixed on typical 8051 devices may be globally disabled, providing power savings capabilities.

The Digital Crossbar allows mapping of internal digital system resources to Port I/O pins (See Figure 1.13). On-chip counter/timers, serial buses, HW interrupts, comparator output, and other digital signals in the controller can be configured to appear on the Port I/O pins specified in the Crossbar Control registers. This allows the user to select the exact mix of general purpose Port I/O and digital resources needed for the particular application.



Figure 1.13. Digital Crossbar Diagram



1.7. 10-Bit Analog to Digital Converter

The C8051F310/1/2/3/6 devices include an on-chip 10-bit SAR ADC with a 25-channel differential input multiplexer. With a maximum throughput of 200 ksps, the ADC offers true 10-bit accuracy with an INL of \pm 1LSB. The ADC system includes a configurable analog multiplexer that selects both positive and negative ADC inputs. Ports1-3 are available as an ADC inputs; additionally, the on-chip Temperature Sensor output and the power supply voltage (V_{DD}) are available as ADC inputs. User firmware may shut down the ADC to save power.

Conversions can be started in six ways: a software command, an overflow of Timer 0, 1, 2, or 3, or an external convert start signal. This flexibility allows the start of conversion to be triggered by software events, a periodic signal (timer overflows), or external HW signals. Conversion completions are indicated by a status bit and an interrupt (if enabled). The resulting 10-bit data word is latched into the ADC data SFRs upon completion of a conversion.

Window compare registers for the ADC data can be configured to interrupt the controller when ADC data is either within or outside of a specified range. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within/outside the specified range.



Figure 1.15. 10-Bit ADC Block Diagram



1.8. Comparators

C8051F31x devices include two on-chip voltage comparators that are enabled/disabled and configured via user software. Port I/O pins may be configured as comparator inputs via a selection mux. Two comparator outputs may be routed to a Port pin if desired: a latched output and/or an unlatched (asynchronous) output. Comparator response time is programmable, allowing the user to select between high-speed and low-power modes. Positive and negative hysteresis are also configurable.

Comparator interrupts may be generated on rising, falling, or both edges. When in IDLE mode, these interrupts may be used as a "wake-up" source. Comparator0 may also be configured as a reset source. Figure 1.16 shows he Comparator0 block diagram.



Figure 1.16. Comparator0 Block Diagram



5.4.2. Window Detector In Differential Mode

Figure 5.8 shows two example window comparisons for right-justified, differential data, with ADC0LTH:ADC0LTL = 0x0040 (+64d) and ADC0GTH:ADC0GTH = 0xFFFF (-1d). In differential mode, the measurable voltage between the input pins is between -VREF and VREF*(511/512). Output codes are represented as 10-bit 2's complement signed integers. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0xFFFF (-1d) < ADC0H:ADC0L < 0x0040 (64d)). In the right example, an AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0xFFFF (-1d) or ADC0H:ADC0L > 0x0040 (+64d)). Figure 5.9 shows an example using left-justified data with the same comparison values.



Figure 5.8. ADC Window Compare Example: Right-Justified Differential Data



Figure 5.9. ADC Window Compare Example: Left-Justified Differential Data





Figure 7.3. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPTnCN (for n = 0 or 1). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in SFR Definition 7.1 and SFR Definition 7.4). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Table 7.1, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CPnHYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see **Section "8.3. Interrupt Handler" on page 93**). The CPnFIF flag is set to logic 1 upon a Comparator falling-edge interrupt, and the CPnRIF flag is set to logic 1 upon the Comparator rising-edge interrupt. Once set, these bits remain set until cleared by software. The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.

The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered-on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed. This Power Up Time is specified in Table 7.1 on page 78.



R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value	
CP1EN	I CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN(00000000	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:	
								0x9A	
Bit7:	CP1EN: Comparator1 Enable Bit.								
	0: Comparator1 Disabled.								
	1: Comparat	or1 Enable	d.						
Bit6:	CP1OUT: Co	omparator1	Output Sta	te Flag.					
	0: Voltage or	n CP1+ < C	P1–.						
	1: Voltage or	1 CP1+ > C	P1–.						
Bit5:	CP1RIF: Co	mparator1 I	Rising-Edge	e Interrupt F	lag.				
	0: No Compa	arator1 Risi	ng Edge In	terrupt has	occurred sir	nce this flag	was last o	cleared.	
D:44	1: Comparat	or'i Rising I	age interr	upt has occ	urrea.				
BIt4:	CP1FIF: Comparator1 Falling-Edge Interrupt Flag.								
	0: No Comparator1 Falling-Edge Interrupt has occurred since this flag was last cleared.								
Dito 2 2	1: Comparator1 Falling-Edge Interrupt has occurred.								
DII53-2.	00: Positivo Hystorosis Disabled								
	UU: POSITIVE HYSTERESIS DISADIED.								
	10: Positive	Hvetorosis	– 10 m\/						
	10. I Ositive	Hysteresis :	= 10 mV. = 20 mV						
Bits1–0 [.]	· · · · · · · · · · · · · · · · · · ·								
Bitor 0.	00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV.								
	10: Negative Hysteresis = 10 mV.								
	11: Negative	Hysteresis	= 20 mV.						
	5	-							

SFR Definition 7.4. CPT1CN: Comparator1 Control



8.3.2. External Interrupts

The /INT0 and /INT1 external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL (/INT0 Polarity) and IN1PL (/INT1 Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (**Section "17.1. Timer 0 and Timer 1" on page 187**) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	/INT0 Interrupt					
1	0	Active low, edge sensitive					
1	1	Active high, edge sensitive					
0	0	Active low, level sensitive					
0	1	Active high, level sensitive					

IT1	IN1PL	/INT1 Interrupt				
1	0	Active low, edge sensitive				
1	1	Active high, edge sensitive				
0	0	Active low, level sensitive				
0	1	Active high, level sensitive				

/INT0 and /INT1 are assigned to Port pins as defined in the IT01CF register (see SFR Definition 8.11). Note that /INT0 and /INT0 Port pin assignments are independent of any Crossbar assignments. /INT0 and /INT1 will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to /INT0 and/or /INT1, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see **Section "13.1. Priority Crossbar Decoder" on page 131** for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

8.3.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 8.4.

8.3.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.



NOTES:



13.1. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 13.3) assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which is always at pins 4 and 5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.3 and/or P0.2 if the external oscillator circuit is enabled, P0.6 if the ADC is configured to use the external conversion start signal (CNVSTR), and any selected ADC or Comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 13.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP, P2SKIP = 0x00); Figure 13.4 shows the Crossbar Decoder priority with the XTAL1 (P0.2) and XTAL2 (P0.3) pins skipped (P0SKIP = 0x0C to skip P0.2 and P0.3 for XTAL use).



Note: P1.6,P1.7,P2.6,P2.7 only available on the C8051F310/1/2/3/4/5; P1SKIP[7:6] should always be set to 11b for the C8051F316/7 devices.

Figure 13.3. Crossbar Priority Decoder with No Pins Skipped



13.2. Port I/O Initialization

Port I/O initialization consists of the following steps:

- Step 1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
- Step 2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
- Step 3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
- Step 4. Assign Port pins to desired peripherals.
- Step 5. Enable the Crossbar (XBARE = (1')).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a '1' indicates a digital input, and a '0' indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 13.4 for the PnMDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMD-OUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR1 is '0', a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a '0' to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to '1' enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility of the Silicon Labs IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. Port output drivers are disabled while the Crossbar is disabled.



	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	SFR Address:
							(bi	t addressable)) 0x80
 Bits7–0: 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 '1' if selected as analog input in register P0MDIN. Directly reads pin when configured as digital input. 0: P0.n pin is logic low. 1: P0.n pin is logic high. 									reads Port

SFR Definition 13.3. P0: Port0

SFR Definition 13.4. POMDIN: Port0 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:	
								0xF1	
Bits7–0:	 Analog Input Configuration Bits for P0.7–P0.0 (respectively). Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled. 0: Corresponding P0.n pin is configured as an analog input. 1: Corresponding P0.n pin is not configured as an analog input. 								



14.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames; however, note that the interrupt is generated before the ACK cycle when operating as a receiver, and after the ACK cycle when operating as a transmitter.

14.5.1. Master Transmitter Mode

Serial data is transmitted on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 14.5 shows a typical Master Transmitter sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that the 'data byte transferred' interrupts occur **after** the ACK cycle in this mode.



Figure 14.5. Typical Master Transmitter Sequence



14.5.2. Master Receiver Mode

Serial data is received on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. After each byte is received, ACKRQ is set to '1' and an interrupt is generated. Software must write the ACK bit (SMB0CN.1) to define the outgoing acknowledge value (Note: writing a '1' to the ACK bit generates an ACK; writing a '0' generates a NACK). Software should write a '0' to the ACK bit after the last byte is received, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. Note that the interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 14.6 shows a typical Master Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur **before** the ACK cycle in this mode.



Figure 14.6. Typical Master Receiver Sequence





Figure 16.2. Multiple-Master Mode Connection Diagram



Figure 16.3. 3-Wire Single Master and Slave Mode Connection Diagram



Figure 16.4. 4-Wire Single Master and Slave Mode Connection Diagram





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



Figure 16.10. SPI Slave Timing (CKPHA = 0)

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

Figure 16.11. SPI Slave Timing (CKPHA = 1)



18. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and five 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled (See Section "13.1. Priority Crossbar Decoder" on page 131 for details on configuring the Crossbar). The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each mode is described in Section "18.2. Capture/Compare Modules" on page 205). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 18.1

Important Note: The PCA Module 4 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. Access to certain PCA registers is restricted while WDT mode is enabled. See **Section 18.3** for details.



Figure 18.1. PCA Block Diagram



18.3. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 4. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH4) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 4 operates as a watchdog timer (WDT). The Module 4 high byte is compared to the PCA counter high byte; the Module 4 low byte holds the offset to be used when WDT updates are performed. The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.

18.3.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2-CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 4 is forced into software timer mode.
- Writes to the Module 4 mode register (PCA0CPM4) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH4 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH4. Upon a PCA0CPH4 write, PCA0H plus the offset held in PCA0CPL4 is loaded into PCA0CPH4 (See Figure 18.10).



Figure 18.10. PCA Module 4 with Watchdog Timer Enabled



NOTES:

