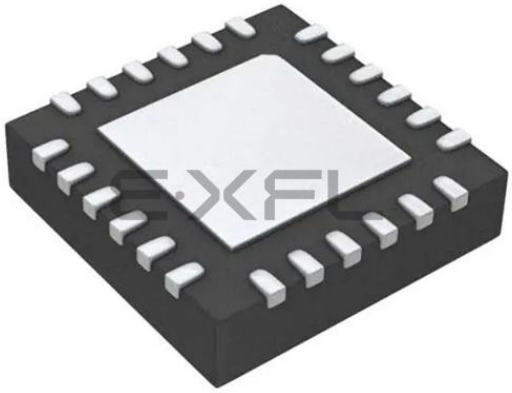


Welcome to E-XFL.COM

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	CIP-51™
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C/SMBus, I²C Slave, SPI, UART/USART
Peripherals	CapSense, DMA, POR, PWM, WDT
Number of I/O	19
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f975-a-gm

2.1.2.1. Normal Mode

Normal mode encompasses the typical full-speed operation. The power consumption of the device in this mode will vary depending on the system clock speed and any analog peripherals that are enabled.

2.1.2.2. Idle Mode

Setting the IDLE bit in PCON causes the hardware to halt the CPU and enter idle mode as soon as the instruction that sets the bit completes execution. 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 is asserted or a reset occurs. The assertion of an enabled interrupt will cause the IDLE bit 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.

2.1.2.3. Stop Mode

Setting the Stop Mode Select bit in the PCON register causes the core to enter stop mode as soon as the instruction that sets the bit completes execution. In Stop mode, the precision internal oscillator and CPU are stopped; the state of the low power oscillator and the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may 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.

2.1.3. Suspend Mode

Setting the Suspend Mode Select bit in the PMU0CF register causes the system clock to be gated off and all internal oscillators disabled. All digital logic (timers, communication peripherals, interrupts, CPU, etc.) stops functioning until one of the enabled wake-up sources occurs.

The following wake-up sources can be configured to wake the device from suspend mode:

- SmaRTClock oscillator fail
- SmaRTClock alarm
- Port Match event
- I2C0 address match
- CS0 end-of-conversion or end-of-scan

2.1.4. Sleep Mode

Setting the Sleep Mode Select bit in the PMU0CF register turns off the internal 1.8 V regulator (VREG0) and switches the power supply of all on-chip RAM to the VDD pin. Power to most digital logic on the chip is disconnected; only PMU0 and the SmaRTClock remain powered. All analog peripherals (ADC0, External Oscillator, etc.) should be disabled prior to entering sleep mode.

RAM and SFR register contents are preserved in sleep mode as long as the voltage on VDD does not fall below VPOR. The PC counter and all other volatile state information is preserved allowing the device to resume code execution upon waking up from sleep mode. The following wake-up sources can be configured to wake the device from sleep mode:

- SmaRTClock oscillator fail
- SmaRTClock alarm
- Port Match event
- I2C0 address match

Table 4.1. Product Selection Guide

Ordering Part Number	Flash Memory (kB)	RAM (Bytes)	Digital Port I/Os (Total)	Number of ADC0 Channels	Pb-Free (RoHS Compliant)	Temperature Range	Package
C8051F970-A-GM	32	8kB	43	43	✓	–40 to 85 °C	QFN-48
C8051F971-A-GM	32	8kB	28	28	✓	–40 to 85 °C	QFN-32
C8051F972-A-GM	32	8kB	19	19	✓	–40 to 85 °C	QFN-24
C8051F973-A-GM	16	4kB	43	43	✓	–40 to 85 °C	QFN-48
C8051F974-A-GM	16	4kB	28	28	✓	–40 to 85 °C	QFN-32
C8051F975-A-GM	16	4kB	19	19	✓	–40 to 85 °C	QFN-24

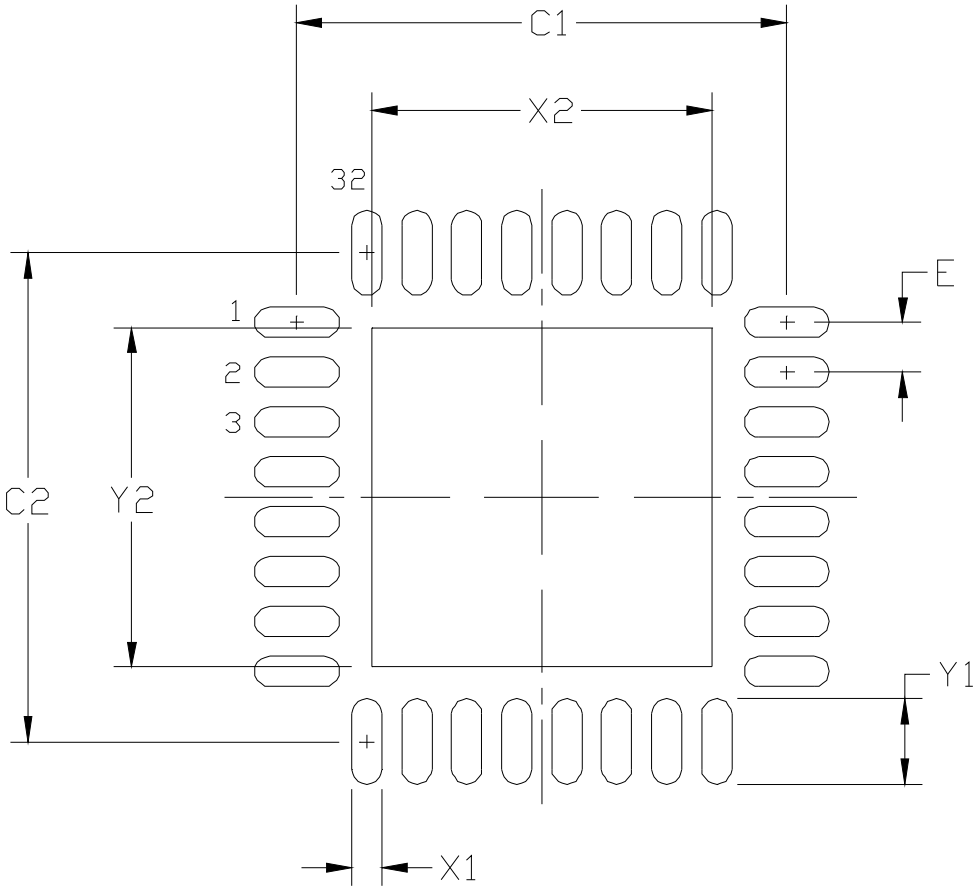


Figure 6.2. Typical QFN-32 Landing Diagram

Table 6.2. QFN-32 PCB Land Pattern

Dimension	MIN	MAX
C1	4.80	4.90
C2	4.80	4.90
E	0.50 BSC	
X1	0.20	0.30
X2	3.20	3.40
Y1	0.75	0.85
Y2	3.20	3.40

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 m minimum, all the way around the pad.

Stencil Design

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
7. A 3 x 3 array of 1.0 mm square openings on 1.2 mm pitch should be used for the center ground pad.

Card Assembly

8. A No-Clean, Type-3 solder paste is recommended.
9. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

Table 7.2. QFN-24 PCB Land Pattern

Dimension	MIN	MAX
C1	3.90	4.00
C2	3.90	4.00
E	0.50 BSC	
X1	0.20	0.30
X2	2.70	2.80
Y1	0.65	0.75
Y2	2.70	2.80

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 µm minimum, all the way around the pad.

Stencil Design

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
7. A 2x2 array of 1.10 mm x 1.10 mm openings on 1.30 mm pitch should be used for the center ground pad.

Card Assembly

8. A No-Clean, Type-3 solder paste is recommended.
9. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

Register 13.4. EIP1: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PT3	PDMA0	PDMA0M	PPCA0	PADC0	PWADC0	PRTC0A	PSMB0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xF6								

Table 13.5. EIP1 Register Bit Descriptions

Bit	Name	Function
7	PT3	**Timer 3 Interrupt Priority Control.** This bit sets the priority of the Timer 3 interrupt. 0: Timer 3 interrupts set to low priority level. 1: Timer 3 interrupts set to high priority level.
6	PDMA0	**DMA0 Interrupt Priority Control.** This bit sets the priority of the DMA0 interrupt. 0: DMA0 interrupts set to low priority level. 1: DMA0 interrupts set to high priority level.
5	PDMA0M	**DMA0 Mid-Point Interrupt Priority Control.** This bit sets the masking of the DMA0 Mid-Point interrupt. 0: Mid-point DMA0 interrupts set to low priority level. 1: Mid-point DMA0 interrupts set to high priority level.
4	PPCA0	**Programmable Counter Array (PCA0) Interrupt Priority Control.** This bit sets the priority of the PCA0 interrupt. 0: PCA0 interrupt set to low priority level. 1: PCA0 interrupt set to high priority level.
3	PADC0	**ADC0 Conversion Complete Interrupt Priority Control.** This bit sets the priority of the ADC0 Conversion Complete interrupt. 0: ADC0 Conversion Complete interrupt set to low priority level. 1: ADC0 Conversion Complete interrupt set to high priority level.
2	PWADC0	**ADC0 Window Comparator Interrupt Priority Control.** This bit sets the priority of the ADC0 Window interrupt. 0: ADC0 Window interrupt set to low priority level. 1: ADC0 Window interrupt set to high priority level.
1	PRTC0A	**RTC Alarm Interrupt Priority Control.** This bit sets the priority of the RTC Alarm interrupt. 0: RTC Alarm interrupt set to low priority level. 1: RTC Alarm interrupt set to high priority level.

RAM and SFR register contents are preserved in sleep mode as long as the voltage on V_{DD} does not fall below VPOR. The PC counter and all other volatile state information is preserved allowing the device to resume code execution upon waking up from sleep mode. The following wake-up sources can be configured to wake the device from sleep mode:

- SmaRTClock Oscillator Fail
- SmaRTClock Alarm
- Port Match Event
- I2C0 Address Match

The V_{DD} supply monitor can be disabled to save power by writing 1 to the MONDIS (PMU0MD.5) bit. When the V_{DD} supply monitor is disabled, all reset sources will trigger a full POR and will re-enable the V_{DD} supply monitor.

In addition, any falling edge on $\overline{\text{RST}}$ (due to a pin reset or a noise glitch) will cause the device to exit sleep mode. In order for the MCU to respond to the pin reset event, software must not place the device back into sleep mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the $\overline{\text{RST}}$ pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into sleep mode. A 4.7 kΩ pullup resistor to V_{DD} is recommend for $\overline{\text{RST}}$ to prevent noise glitches from waking the device.

Note: Entering Sleep mode may cause the MCU to disconnect from the debug adapter while debugging.

16.6. Low Power Active Mode

Running in normal active mode can waste a significant amount of amount of power by clocking unused peripherals. Low power active mode in C8051F97x devices allows control of clocking activity in the clock tree, which enables firmware to shut off clocking to unused peripherals and save power.

Setting bit 1 and 2 of CLKMODE register causes the CIP-51 to enter low power active mode as soon as the instruction that sets the bits completes execution. CPU, all the analog and digital peripherals remain active except those whose clocks are turned off by user. **The CPU will not be able to access SFR of peripherals on inactive branches of the clock tree**. Refer to Figure 16.1 for more information on how modules receive or request for clocks in low power modes. **It is important that firmware configures bit 4 to 7 of PCLKEN register on page 248 to ensure desired clock gating of peripherals during low power active mode**. For example, if UART is supposed to be active during low power active mode, according to the PCLKEN register description bit 4 should be set. However, that is not sufficient in this case because Timer 1 is needed for UART Baud rate generation. As a consequence, bit 7 should be set as well for proper UART operation.

Low power active mode is terminated when the CLKMODE register is programmed to 0x00 or a reset occurs. Systems that use all the peripherals and always stay in the active mode may not find improvement in power consumption in the low power active mode due overhead logic required for this implementation.

16.7. Low Power Idle Mode

In this mode of operation, the CPU is halted and clocks supplied to unused peripherals can be shut down to save power. Clocks of these peripherals can be enabled or disabled by programming bits 0 to 3 of PCLKEN register accordingly on page 248. The device should be put to low power mode by setting bits 1 and 2 of CLKMODE register. The last step necessary to put device in low power idle mode is to set the Idle Mode Select bit (PCON.0) to 1. The CPU will be halted and the device will enter low-power idle mode as soon as the instruction that sets the Idle Mode Select bit completes execution. As a result, configuration of peripherals clock gating through PCLKEN register and CLKMODE register should be properly prepared before the Idle Mode Select bit in PCON is set. All internal registers and memory maintain their original data. All the analog and digital peripherals remain active except those whose clocks are turned off by user. Modules that are capable of requesting for clocks can do so at any time. Refer to Figure 16.1 for more information on how modules receive or request for clocks in low power modes.

Note: To ensure the MCU enters a low power state upon entry into Low Power Idle mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6).

16.10. Power Control Registers

Register 16.1. PCON: Power Control

Bit	7	6	5	4	3	2	1	0
Name	GF5	GF4	GF3	GF2	GF1	GF0	STOP	IDLE
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0x87								

Bit	Name	Function
7	GF5	**General Purpose Flag 5.** This flag is a general purpose flag for use under firmware control.
6	GF4	**General Purpose Flag 4.** This flag is a general purpose flag for use under firmware control.
5	GF3	**General Purpose Flag 3.** This flag is a general purpose flag for use under firmware control.
4	GF2	**General Purpose Flag 2.** This flag is a general purpose flag for use under firmware control.
3	GF1	**General Purpose Flag 1.** This flag is a general purpose flag for use under firmware control.
2	GF0	**General Purpose Flag 0.** This flag is a general purpose flag for use under firmware control.
1	STOP	**Stop Mode Select.** Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0.
0	IDLE	**Idle Mode Select.** Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0.

Register 21.3. DMA0MINT: DMA0 Mid-Point Interrupt Flags

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CH6MI	CH5MI	CH4MI	CH3MI	CH2MI	CH1MI	CH0MI
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x88 (bit-addressable)								

Bit	Name	Function
7	Reserved	Must write reset value.
6	CH6MI	**Channel 6 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 6.
5	CH5MI	**Channel 5 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 5.
4	CH4MI	**Channel 4 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 4.
3	CH3MI	**Channel 3 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 3.
2	CH2MI	**Channel 2 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 2.
1	CH1MI	**Channel 1 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 1.
0	CH0MI	**Channel 0 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 0.

Note: Mid-point Interrupt flag is set when the offset address DMA0NAOH/L equals to half of data transfer size DMA0NSZH/L if the transfer size is an even number or half of data transfer size DMA0NSZH/L plus one if the transfer size is an odd number. Firmware must clear this flag. The mid-point interrupt is enabled by setting the MIEN bit in the DMA0NCF register with DMA0SEL configured for the corresponding channel.

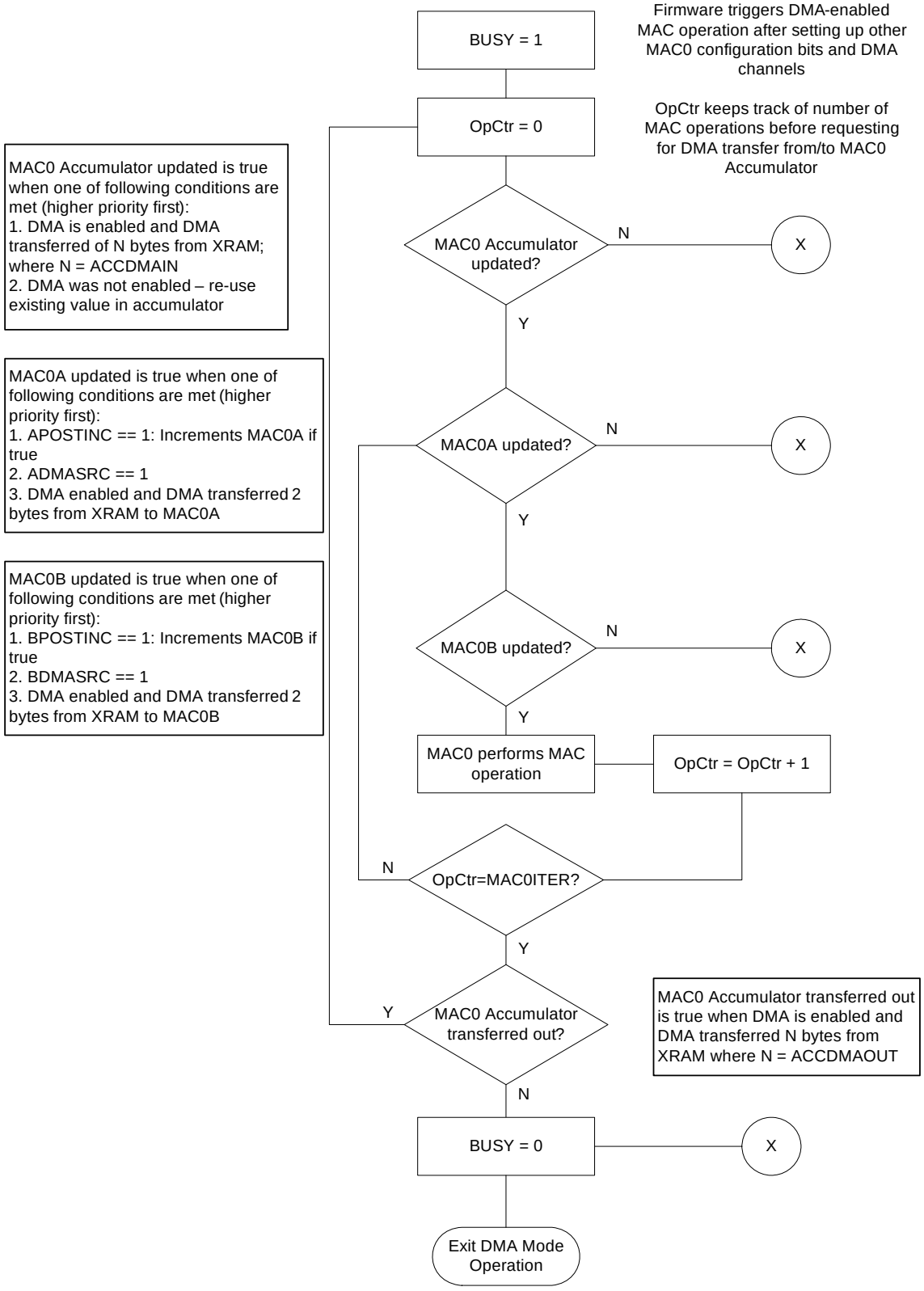
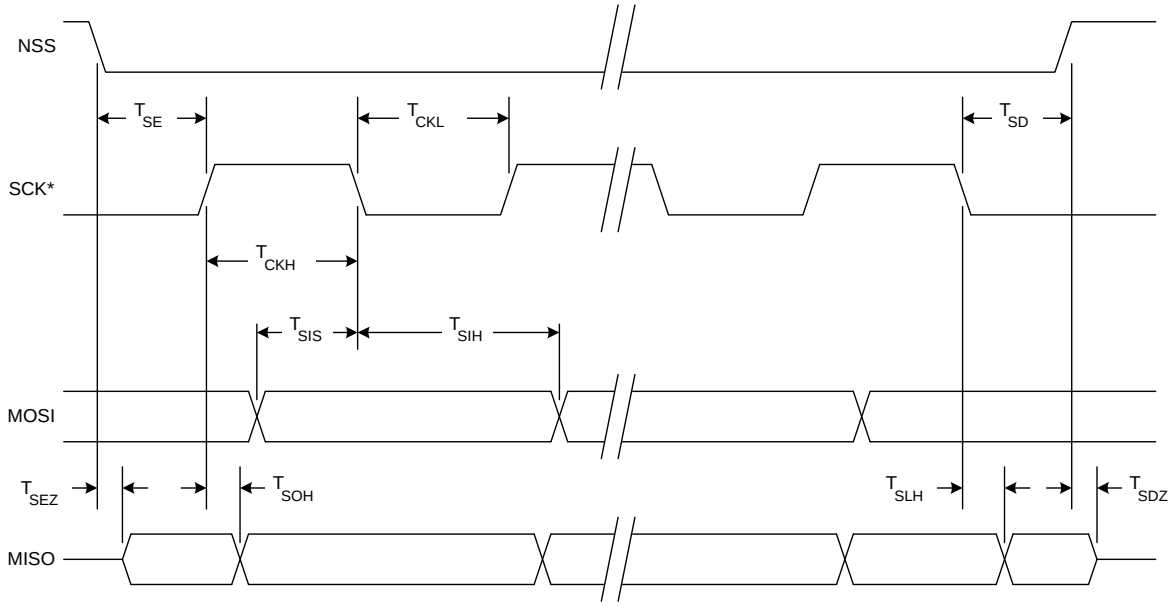


Figure 22.4. DMA Mode Operation Flow Chart



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

Figure 28.11. SPI Slave Timing (CKPHA = 1)

Table 29.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1) (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	0	0	X	A slave address + R/W was received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
		0	1	X	Lost arbitration as master; slave address + R/W received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
						Reschedule failed transfer	1	0	X	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		0	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	0	0	X	A slave byte was received.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	0000
						Set NACK for next data byte; Read SMB0DAT.	0	0	0	0000
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	0	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110

Table 31.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 49 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 Timer1 chapter.
2. X = Don't care.

32.1. Timer 0 and Timer 1

Timer 0 and Timer 1 are each implemented as a16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register. Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently for the operating modes described below.

32.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, CT0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

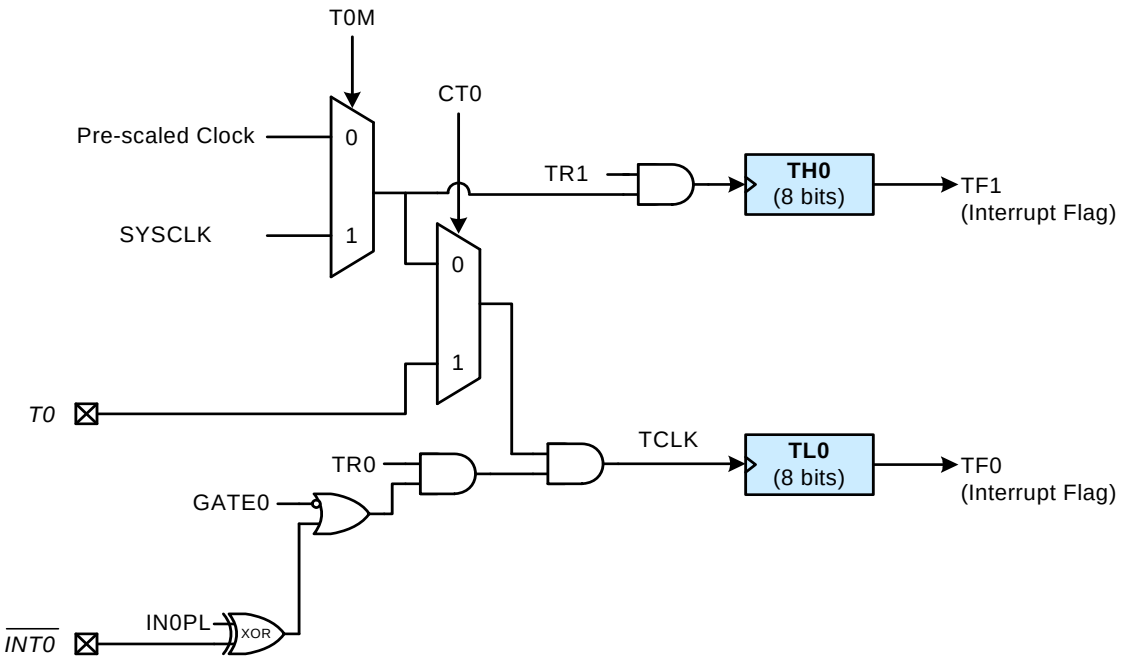


Figure 32.3. T0 Mode 3 Block Diagram

32.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit in TMR2CN defines the Timer 2 operation mode. Timer 2 can also be used in Capture Mode to measure the SmaRTClock or the Comparator 0 period with respect to another oscillator. The ability to measure the Comparator 0 period with respect to the system clock makes using Touch Sense switches very easy.

Timer 2 may be clocked by the system clock, the system clock divided by 12, SmaRTClock divided by 8, or Comparator 0 output. Note that the SmaRTClock divided by 8 and Comparator 0 output is synchronized with the system clock.

32.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT in the TMR2CN register is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, SmaRTClock divided by 8, or Comparator 0 output. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 32.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LINT bit is set, an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

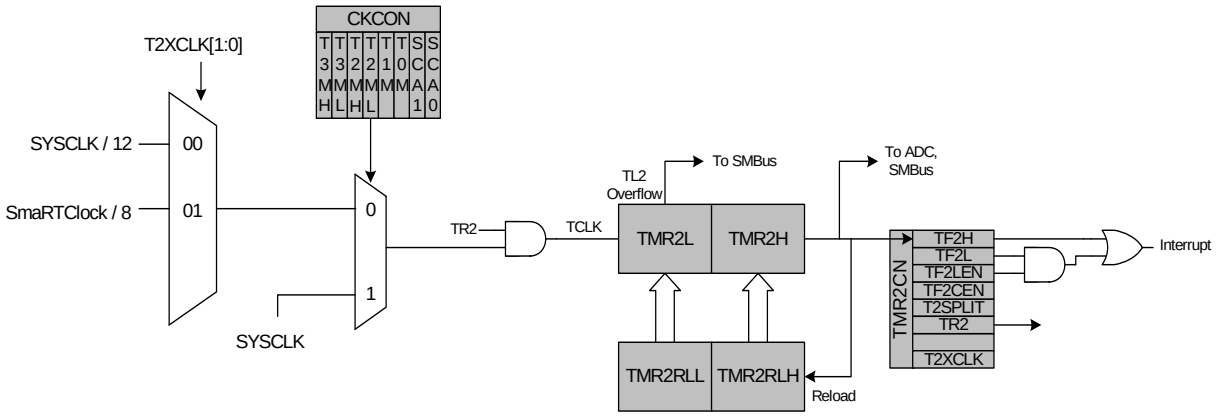


Figure 32.4. Timer 2 16-Bit Mode Block Diagram

Register 32.14. TMR3RLL: Timer 3 Reload Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x92								

Table 32.16. TMR3RLL Register Bit Descriptions

Bit	Name	Function
7:0	TMR3RLL	**Timer 3 Reload Low Byte.** When operating in one of the auto-reload modes, TMR3RLL holds the reload value for the low byte of Timer 3 (TMR3L). When operating in capture mode, TMR3RLL is the captured value of TMR3L.

33.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA 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. An interrupt request is generated if the CCFn interrupt for that module is 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. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

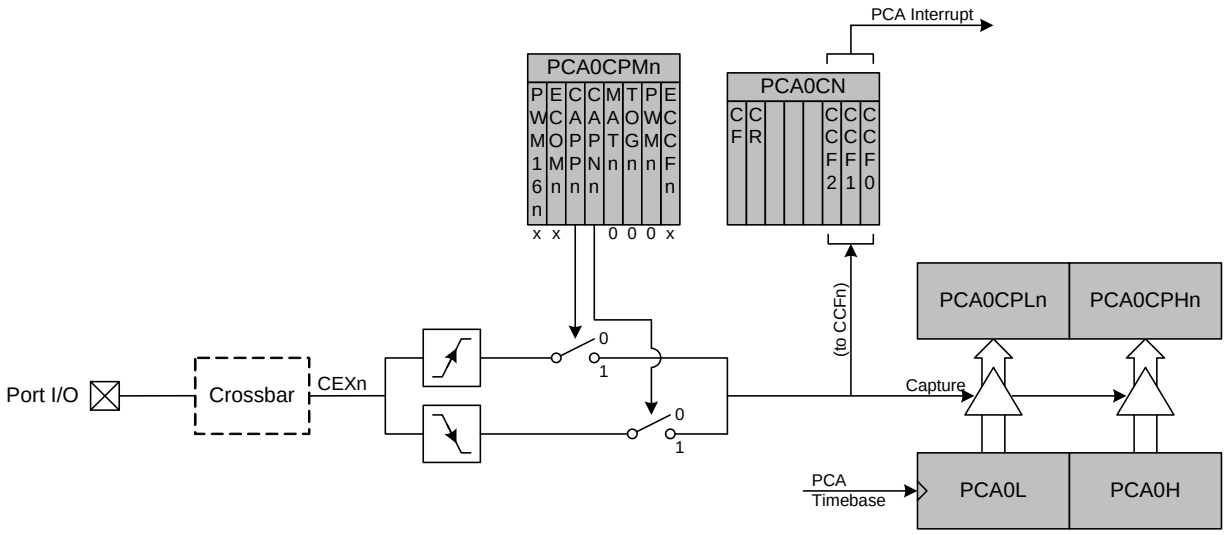


Figure 33.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

33.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 33.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 33.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

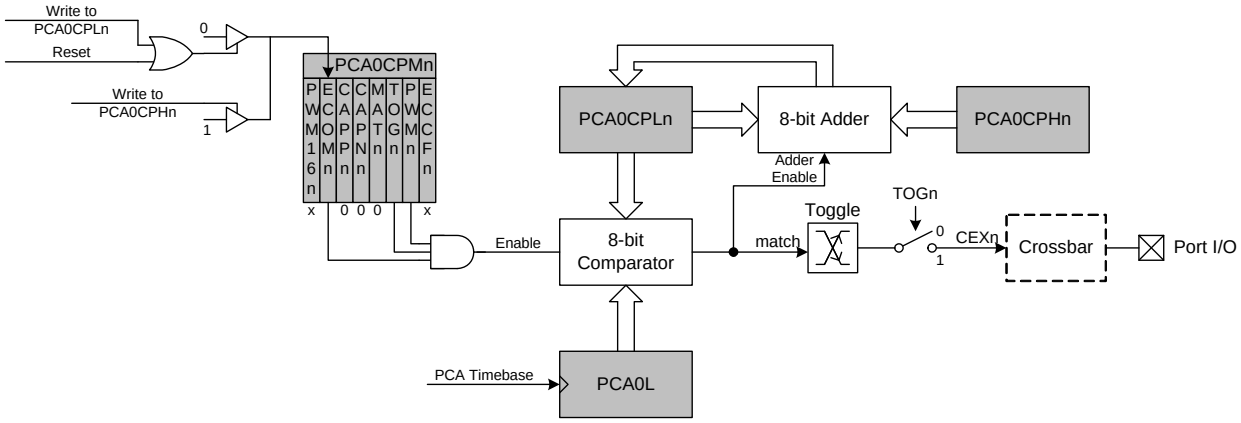


Figure 33.7. PCA Frequency Output Mode

33.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

Register 34.3. C2REVID: C2 Revision ID

Bit	7	6	5	4	3	2	1	0
Name	C2REVID							
Type	R							
Reset	X	X	X	X	X	X	X	X
C2 Address: 0x01								

Table 34.3. C2REVID Register Bit Descriptions

Bit	Name	Function
7:0	C2REVID	**Revision ID.** This read-only register returns the 8-bit revision ID. For example: 0x01 = Revision A.