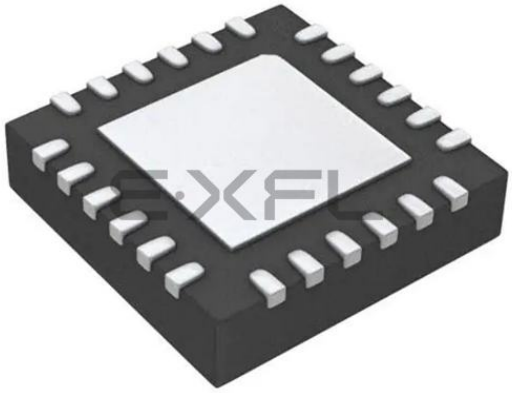

Welcome to E-XFL.COM

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	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K 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/c8051f972-a-gm

6.1. QFN-32 Package Marking

The first row of the package marking is the part number, including the revision. The second row is the 6-digit trace code indicating assembly information. The last row indicates year (two digits) and workweek (two digits) when the device was packaged.

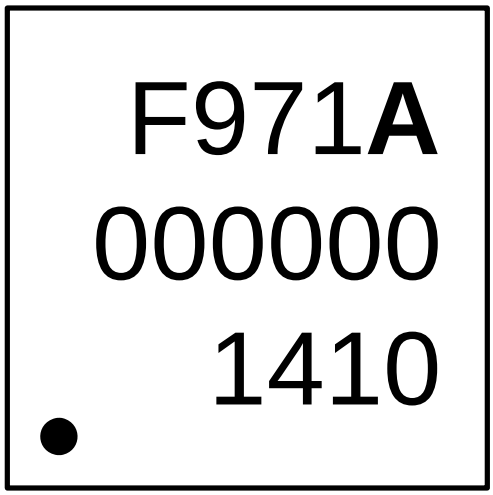

Figure 6.3. QFN-32 Package Marking

Register 13.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3	EDMA0	EDMA0M	EPCA0	EADC0	EWADC0	ERTC0A	ESMB0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xE6								

Table 13.4. EIE1 Register Bit Descriptions

Bit	Name	Function
7	ET3	**Timer 3 Interrupt Enable.** This bit sets the masking of the Timer 3 interrupt. 0: Disable Timer 3 interrupts. 1: Enable interrupt requests generated by the TF3L or TF3H flags.
6	EDMA0	**DMA0 Interrupt Enable.** This bit sets the masking of the DMA0 Interrupt. 0: Disable DMA0 interrupts. 1: Enable interrupt requests generated by DMA0.
5	EDMA0M	**DMA0 Mid-Point Interrupt Enable.** This bit sets the masking of the DMA0 Mid-Point interrupt. 0: Disable DMA0M interrupts. 1: Enable interrupt requests generated by the DMA0M flags.
4	EPCA0	**Programmable Counter Array (PCA0) Interrupt Enable.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3	EADC0	**ADC0 Conversion Complete Interrupt Enable.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the ADINT flag.
2	EWADC0	**Window Comparison ADC0 Interrupt Enable.** This bit sets the masking of ADC0 Window Comparison interrupt. 0: Disable ADC0 Window Comparison interrupt. 1: Enable interrupt requests generated by ADC0 Window Compare flag (ADWINT).
1	ERTC0A	**RTC Alarm Interrupts Enable.** This bit sets the masking of the RTC Alarm interrupt. 0: Disable RTC Alarm interrupts. 1: Enable interrupt requests generated by a RTC Alarm.

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).

18.13. Adjusting CS0 For Special Situations

There are several configuration options in the CS0 module designed to modify the operation of the circuit and address special situations. In particular, any circuit with more than 500 Ω of series impedance between the sensor and the device pin may require the adjustments detailed in this section for optimal performance. Typical applications which may require adjustments include the following:

- Touch panel sensors fabricated using a resistive conductor such as indium-tin-oxide (ITO).
- Circuits using a high-value series resistor to isolate the sensor element for high ESD protection.

Capacitive sensors created using PCB traces should generally require no fine tuning, and the default settings for CS0DT, CS0DR, CS0IA, CS0RP and CS0LP should be used.

18.13.1. Adjusting the CS0 Reset Timing

The CS0 module determines capacitance by discharging an external capacitor and then measuring how quickly that capacitor charges. In order to do this, the external capacitor must be fully discharged before every test. There are two timers inside the CS0 module which determine the timing for the reset (discharge) operation.

CS0 performs a two-stage discharge (double reset) of the external capacitor at the start of every bit conversion to improve performance in high-noise environments. In this method, most of the charge in the external capacitor is removed in a first reset stage through a low-resistance switch to ground. A second reset is then performed using a high-resistance switch to ground. This second reset removes any ambient noise energy that might have been captured in the external capacitor at the end of the first reset stage.

The lengths of both reset periods are independently adjustable. Longer periods are used when the external capacitor is separated from the CS0 module by a large resistor (more than 500 Ω) because that series resistor would slow the rate of discharge.

Determining the appropriate settings for CS0DT (the primary reset) and CS0DR (the secondary reset) are two of a series of related adjustments which must be made when using CS0 to measure capacitive loads in the presence of high resistance.

18.13.2. Adjusting Primary Reset Timing: CS0DT

Primary reset timing adjustment is performed to provide peak sensitivity for highly-resistive loads and peak linearity for capacitive loads linked thorough a distributed resistance (such as an ITO touch panel) while minimizing the required conversion time.

The adjustment for CS0DT should be performed while CS0DR and CS0IA bits are set at their maximum values (CS0DR = 11b, CS0IA = 001b).

1. Begin the adjustment with CS0DT set to maximum delay (CS0DT = 111b). Measure the untouched average CS0 result for the channel under test.

Note: When calibrating CS0 for use with an ITO panel, consider the use of an artificial finger: a small (¼" O.D.) washer (#2 regular, #4 narrow) wired through a 1000 pF capacitor to ground. Select a touch point at the on the farthest end of the longest row. Find the point where maximum response is returned from the CS0 conversion.

2. Record the average touched CS0 value with CS0DT = 111b. The touched value should be higher than the untouched value. The magnitude of the difference between the touched and untouched average CS0 values is the figure of merit for touch sensitivity.
3. Decrease the primary reset time CS0DT by one and repeat the touched and untouched CS0 measurements. Repeat this step until values have been recorded for all eight CS0DT settings. As the CS0DT setting decreases, the average sensitivity of the CS0 value may begin to decrease significantly.
4. CS0DT should be set high enough that there is not a significant decrease in sensitivity due to resistance. Select the CS0DT setting that occurred prior to the observed drop in CS0 touch sensitivity.

Table 20.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1

Register 20.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS		OV	F1	PARITY
Type	RW	RW	RW	RW		RW	RW	R
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xD0 (bit-addressable)								

Table 20.7. PSW Register Bit Descriptions

Bit	Name	Function
7	CY	**Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	**Auxiliary Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under firmware control.
4:3	RS	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: 1. An ADD, ADDC, or SUBB instruction causes a sign-change overflow. 2. A MUL instruction results in an overflow (result is greater than 255). 3. A DIV instruction causes a divide-by-zero condition. The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under firmware control.
0	PARITY	**Parity Flag.** This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

When the FRACMD bit is set to 1, the inputs are treated at 16-bit fractional values. The decimal point is located between bits 15 and 14 of the data word. After the operation, the accumulator will contain a 40-bit fractional value, with the decimal point located between bits 31 and 30. Figure 22.3 shows how fractional numbers are stored in the SFRs.

MAC0A and MAC0B Bit Weighting

	High Byte							Low Byte							
s*1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

If (MAC0SN == 1), s=-1, otherwise s=1

MAC0 Accumulator Bit Weighting (no rounding)

MAC0OVR				MAC0ACC3 : MAC0ACC2 : MAC0ACC1 : MAC0ACC0										
$s*2^{8}$	2^{7}		2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}		2^{-27}	2^{-28}	2^{-29}	2^{-30}	2^{-31}	

s = sign of result (1 or -1)

MAC0 Accumulator rounded result (MAC0RND=1) Bit Weighting

	MAC0ACC1							MAC0ACC0							
s*1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

s = sign of result (1 or -1)

Figure 22.3. Fractional Mode Data Representation

22.3. Operating in Multiply and Accumulate Mode

MAC0 operates in multiply and accumulate (MAC) mode when the ACCMD bit (MAC0CF0.3) is cleared to 0. When operating in MAC mode, MAC0 performs a 16-by-16 bit multiply on the contents of the MAC0A and MAC0B registers and adds the result to the contents of the 41-bit accumulator. A MAC operation takes 1 SYSCLK cycle to complete. A rounded (and optionally, saturated) result is available when the MAC0RND bit is set.

If the CLRACC bit (MAC0CF0.5) is set to 1, the accumulator and all MAC0STA flags will be cleared during the next SYSCLK cycle. The CLRACC bit will clear itself to 0 when the clear operation has completed.

22.4. Operating in Multiply Only Mode

MAC0 operates in multiply only mode when the ACCMD bit (MAC0CF0.5) is set to 1. Multiply only mode is identical to Multiply and Accumulate mode, except that the multiplication result is added with a value of zero before being stored in the 41-bit accumulator (i.e. it overwrites the current accumulator contents). As in MAC mode, the rounded result can be read if the ROUND bit is set. Note that in Multiply Only mode, the HOVF flag is not affected.

24. Clocking Sources

The C8051F97x devices can be clocked from the internal low-power 24.5 MHz oscillator, the internal low-power 20 MHz oscillator, the SmaRTClock real time clock oscillator, or externally by an external oscillator (not supported on devices in the QFN-24 package). An adjustable clock divider allows the selected clock source to be post-scaled by powers of 2, up to a factor of 128. By default, the system clock comes up as the 24.5 MHz oscillator divided by 8.

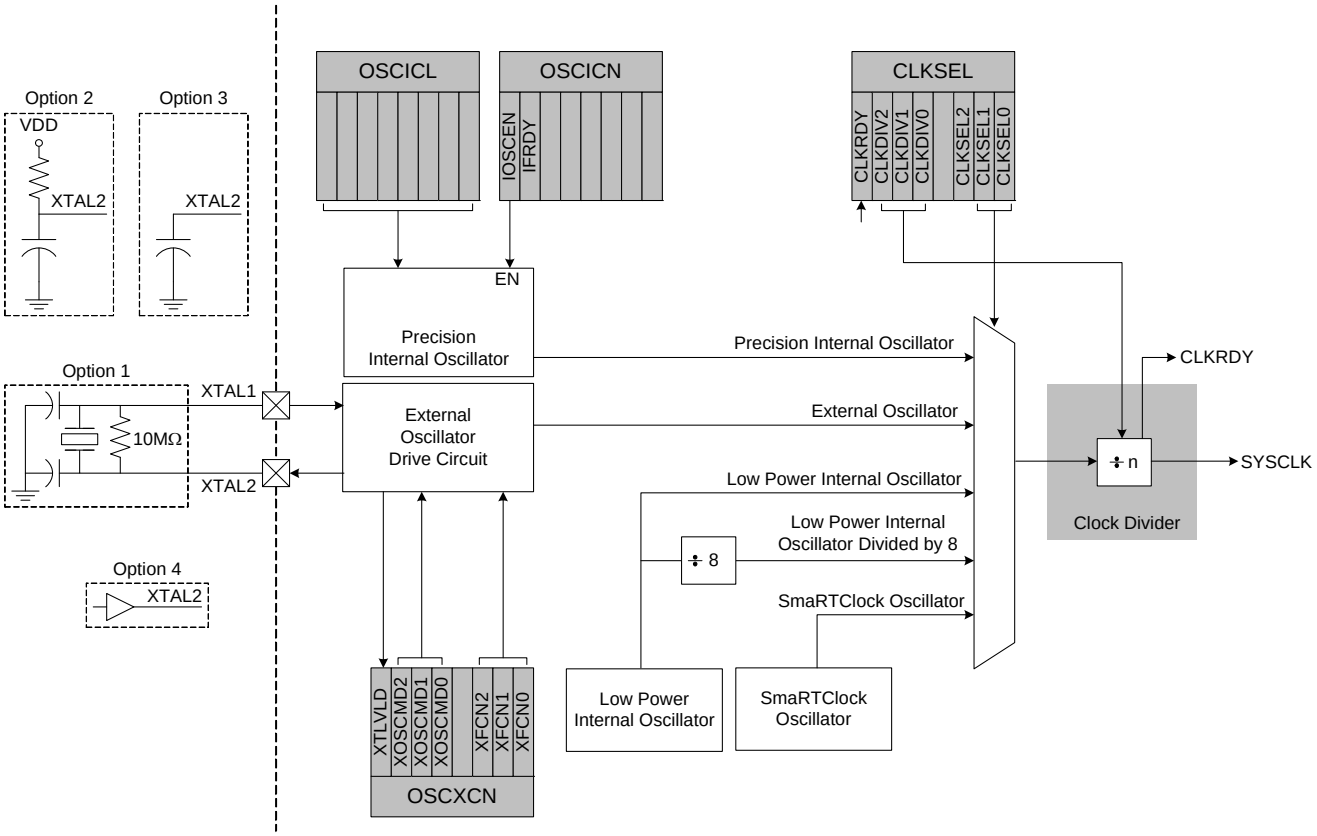

Figure 24.1. Clocking Sources Block Diagram

The proper way of changing the system clock when both the clock source and the clock divide value are being changed is as follows:

If switching from a fast “undivided” clock to a slower “undivided” clock:

1. Change the clock divide value.
2. Poll for CLKRDY > 1.
3. Change the clock source.

If switching from a slow “undivided” clock to a faster “undivided” clock:

1. Change the clock source.
2. Change the clock divide value.
3. Poll for CLKRDY > 1.

Register 24.2. PCLKEN: Low Power Peripheral Clock Enable

Bit	7	6	5	4	3	2	1	0
Name	PCLKEN7	PCLKEN6	PCLKEN5	PCLKEN4	PCLKEN3	PCLKEN2	PCLKEN1	PCLKEN0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xF6								

Bit	Name	Function
7	PCLKEN7	**Low Power Active Mode Peripheral Set 3 Enable.** 0: Disable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power active mode. 1: Enable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power active mode.
6	PCLKEN6	**Low Power Active Mode Peripheral Set 2 Enable.** 0: Disable clocks of PCA0, CS0, and ADC0 in low power active mode. 1: Enable clocks of PCA0, CS0, and ADC0 in low power active mode.
5	PCLKEN5	**Low Power Active Mode Peripheral Set 1 Enable.** 0: Disable clocks of MAC0 in low power active mode. 1: Enable clocks of MAC0 in low power active mode.
4	PCLKEN4	**Low Power Active Mode Peripheral Set 0 Enable.** 0: Disable clocks of UART0, Timer 3, SPI0, and I2C0 in low power active mode. 1: Enable clocks of UART0, Timer 3, SPI0, and I2C0 in low power active mode.
3	PCLKEN3	**Low Power Idle Mode Peripheral Set 3 Enable.** 0: Disable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power idle mode. 1: Enable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power idle mode.
2	PCLKEN2	**Low Power Idle Mode Peripheral Set 2 Enable.** 0: Disable clocks of PCA0, CS0, and ADC0 in low power idle mode. 1: Enable clocks of PCA0, CS0, and ADC0 in low power idle mode.
1	PCLKEN1	**Low Power Idle Mode Peripheral Set 1 Enable.** 0: Disable clocks of MAC0 in low power idle mode. 1: Enable clocks of MAC0 in low power idle mode.
0	PCLKEN0	**Low Power Idle Mode Peripheral Set 0 Enable.** 0: Disable clocks of UART0, Timer 3, SPI0, and I2C0 in low power idle mode. 1: Enable clocks of UART0, Timer 3, SPI0, and I2C0 in low power idle mode.

26.4. Port I/O Modes of Operation

Port pins are configured by firmware as digital or analog I/O using the PnMDIN registers. On reset, all port I/O cells default to a high impedance state with weak pull-ups enabled. Until the crossbar is enabled, both the high and low port I/O drive circuits are explicitly disabled on all crossbar pins. Port pins configured as digital I/O may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

26.4.1. Configuring Port Pins For Analog Modes

Any pins to be used for analog functions should be configured for analog mode. When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. Port pins configured for analog functions will always read back a value of ‘0’ in the corresponding Pn Port Latch register. To configure a pin as analog, the following steps should be taken:

1. Clear the bit associated with the pin in the PnMDIN register to ‘0’. This selects analog mode for the pin.
2. Set the bit associated with the pin in the Pn register to ‘1’.
3. Skip the bit associated with the pin in the PnSKIP register to ensure the crossbar does not attempt to assign a function to the pin.

26.4.2. Configuring Port Pins For Digital Modes

Any pins to be used by digital peripherals or as GPIO should be configured as digital I/O (PnMDIN.n = ‘1’). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = ‘1’) drive the port pad to the supply rails based on the output logic value of the port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the port pad to the low-side rail when the output logic value is ‘0’ and become high impedance inputs (both high low drivers turned off) when the output logic value is ‘1’.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the port pad to the high-side rail to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven low to minimize power consumption, and they may be globally disabled by setting WEAKPUD to ‘1’. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the port pad, regardless of the output logic value of the port pin.

To configure a pin as digital input:

1. Set the bit associated with the pin in the PnMDIN register to ‘1’. This selects digital mode for the pin.
2. Clear the bit associated with the pin in the PnMDOUT register to ‘0’. This configures the pin as open-drain.
3. Set the bit associated with the pin in the Pn register to ‘1’. This tells the output driver to “drive” logic high. Because the pin is configured as open-drain, the high-side driver is not active, and the pin may be used as an input.

Open-drain outputs are configured exactly as digital inputs. However, the pin may be driven low by an assigned peripheral, or by writing ‘0’ to the associated bit in the Pn register if the signal is a GPIO.

To configure a pin as a digital, push-pull output:

1. Set the bit associated with the pin in the PnMDIN register to ‘1’. This selects digital mode for the pin.
2. Set the bit associated with the pin in the PnMDOUT register to ‘1’. This configures the pin as push-pull.

If a digital pin is to be used as a general-purpose I/O, or with a digital function that is not part of the crossbar, the bit associated with the pin in the PnSKIP register can be set to ‘1’ to ensure the crossbar does not attempt to assign a function to the pin.

26.4.3. Port Drive Strength

Port drive strength can be controlled on a pin-by-pin basis using the PnDRV registers. Each pin has a bit in the associated PnDRV register to select the high or low drive strength setting for the pin. By default, all pins are configured for low drive strength.

The SPI0 Clock Rate Register (SPI0CKR) 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 or 12.5 MHz, whichever is slower. 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 slave's 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 slave's system clock.

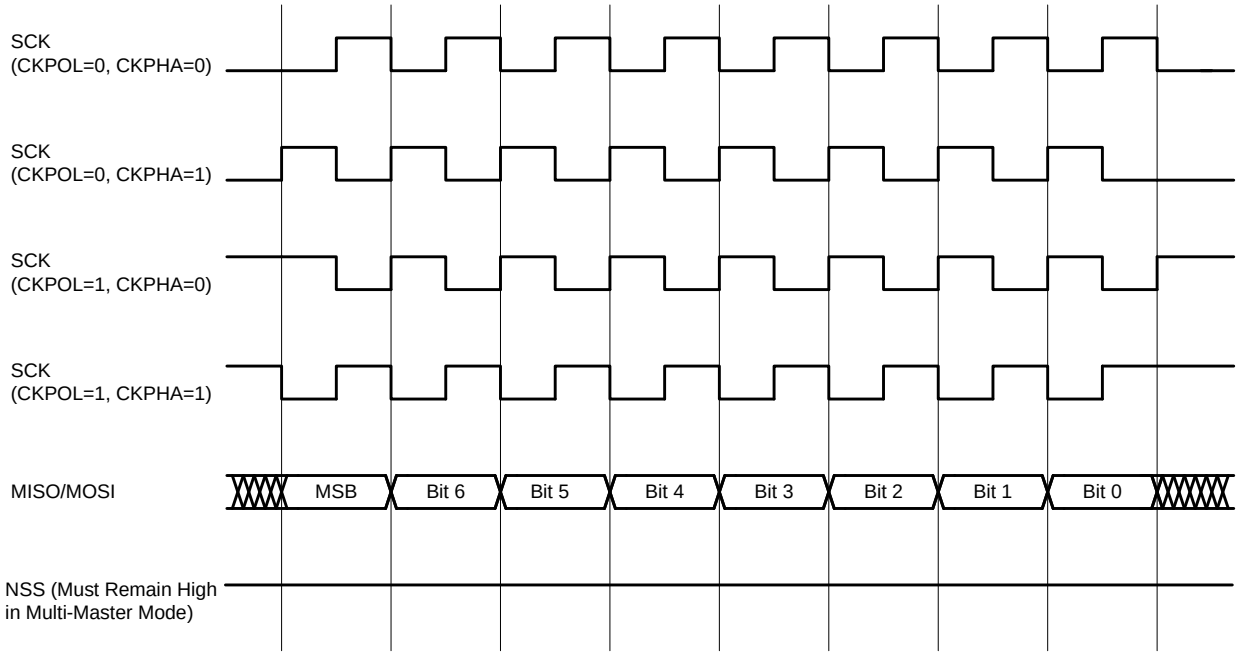

Figure 28.5. Master Mode Data/Clock Timing

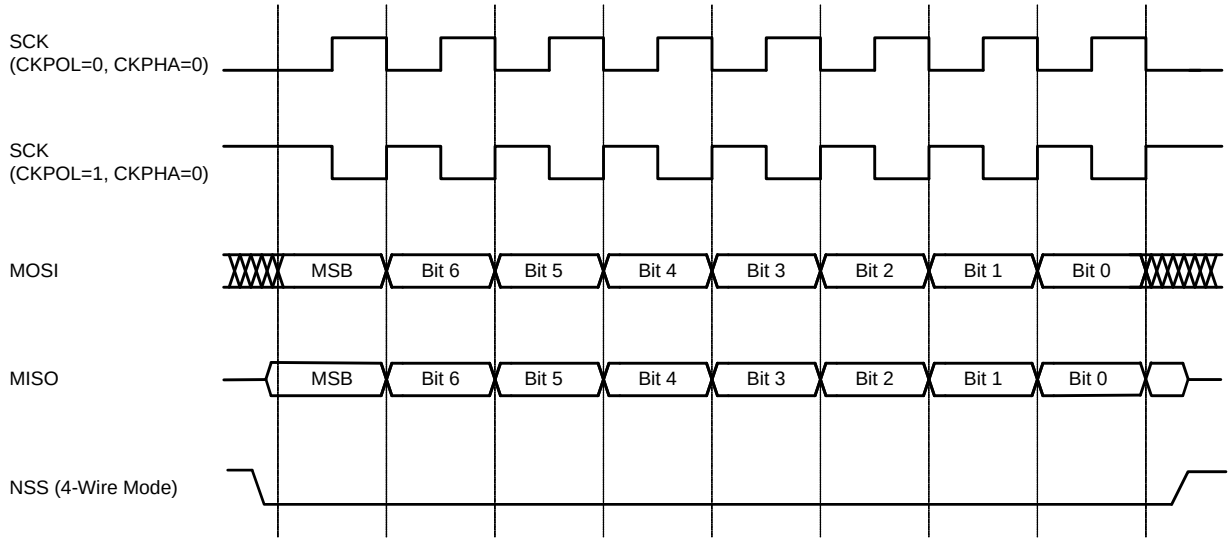

Figure 28.6. Slave Mode Data/Clock Timing (CKPHA = 0)

Table 29.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
Note: Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgment, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI0 is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.		

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “29.3.4. SCL Low Timeout” on page 344). The SMBus interface will force the associated timer to reload while SCL is high, and allow the timer to count when SCL is low. The timer interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 29.4).

29.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information. The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 29.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

29.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

29.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 29.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 29.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 29.5 for SMBus status decoding using the SMB0CN register.

30.1. Supporting Documents

It is assumed that the reader is familiar or has access to the following supporting documents:

- The I²C-bus specification and the user manual Rev. 0.3.

30.2. The I²C Configuration

Figure 30.2 shows a typical I²C configuration. The I²C specification allows any recessive voltage between 3.0 and 5.0 V; different devices on the bus may operate at different voltage levels.

Note: The port pins on the C8051F97x devices are not 5 V tolerant, therefore, the device may only be used in I²C networks where the supply voltage does not exceed V_{DD}.

The bidirectional SCL and SDA lines must be connected to a positive power supply voltage through a pull-up resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed the specifications defined in the I²C standard.

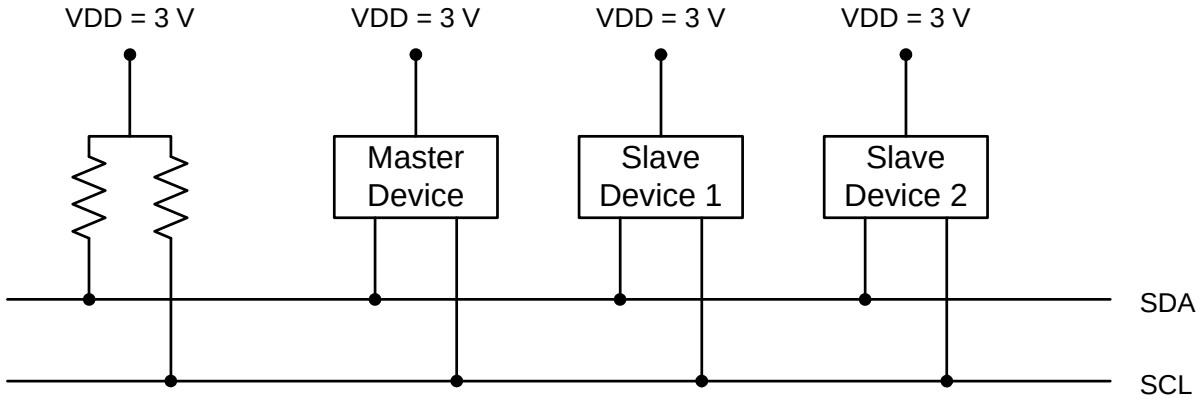

Figure 30.2. Typical I²C Configuration

30.3. I2CSLAVE0 Operation

The I2CSLAVE0 peripheral supports two types of data transfers: I²C Read data transfers where data is transferred from the C8051F97x's I²C slave peripheral to an I²C master, and I²C Write data transfers where data is transferred from an I²C master to the C8051F97x's I²C slave peripheral. The I²C master initiates both types of data transfers and provides the serial clock pulses that the I²C slave peripheral detects on the SCL pin.

A typical I²C transaction consists of a START condition followed by an address byte (Bits7-1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received are acknowledged (ACK) with a low SDA during a high SCL (refer to Figure 30.3).

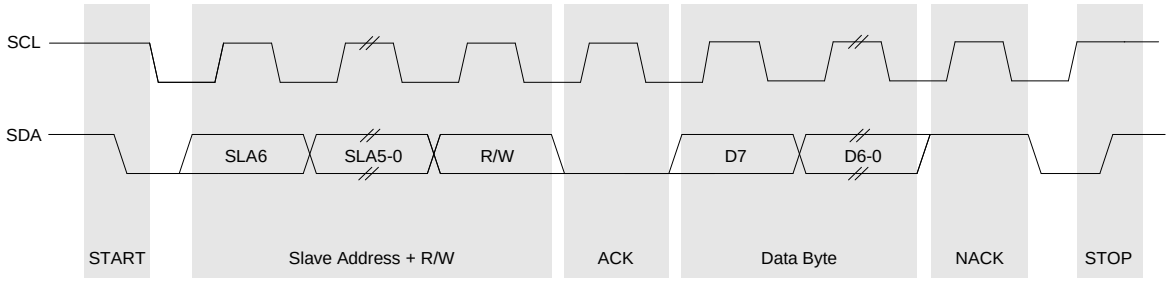

Figure 30.3. I²C Transaction

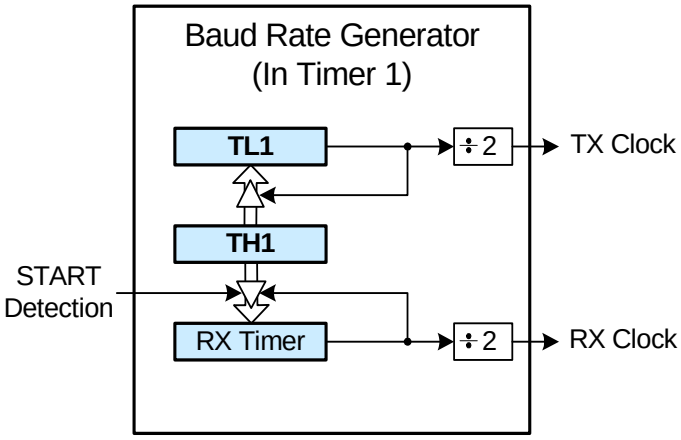

Figure 31.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload. The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK/4, SYSCLK/12, SYSCLK/48, the external oscillator clock/8, or an external input T1. For any given Timer 1 overflow rate, the UART0 baud rate is determined by Equation 31.1.

$$\text{UartBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

Equation 31.1. UART0 Baud Rate

Timer 1 overflow rate is selected as described in the Timer section. A quick reference for typical baud rates and system clock frequencies is given in Table 31.1.

32.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal $\overline{INT0}$ is active as defined by bit IN0PL in register IT01CF.

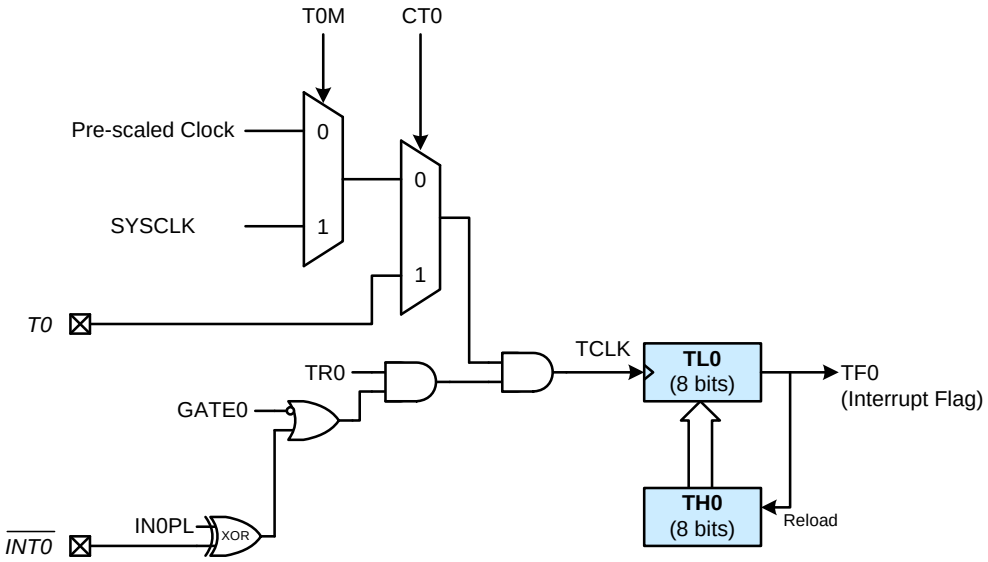

Figure 32.2. T0 Mode 2 Block Diagram

32.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit in the TMR3CN register defines the Timer 3 operation mode. Timer 3 can also be used in Capture Mode to measure the external oscillator source or the SmaRTClock oscillator period with respect to another oscillator.

Timer 3 may be clocked by the system clock, the system clock divided by 12, external oscillator source divided by 8, or the SmaRTClock oscillator. The external oscillator source divided by 8 and SmaRTClock oscillator is synchronized with the system clock.

32.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT in the TMR3CN register is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, external oscillator clock source divided by 8, or SmaRTClock oscillator. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 32.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LINT bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

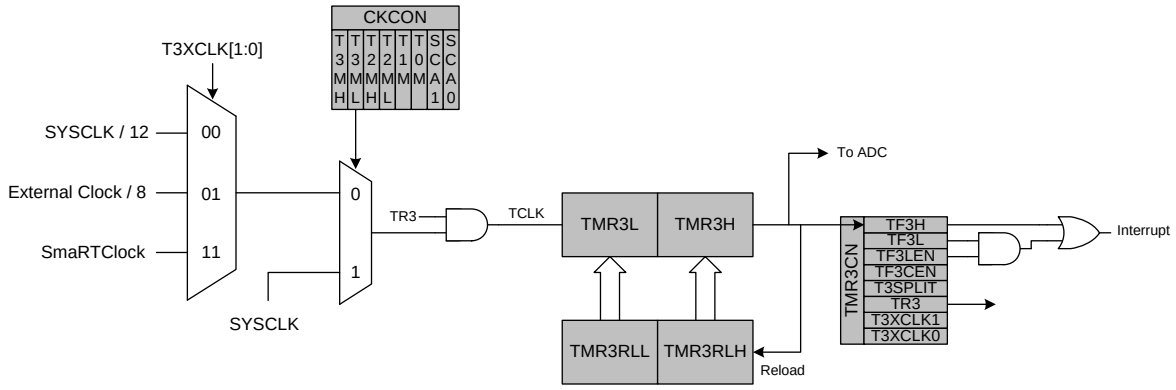

Figure 32.7. Timer 3 16-Bit Mode Block Diagram

Register 32.12. TMR2H: Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xCD								

Table 32.14. TMR2H Register Bit Descriptions

Bit	Name	Function
7:0	TMR2H	**Timer 2 High Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

Table 32.15. TMR3CN Register Bit Descriptions

Bit	Name	Function
1:0	T3XCLK	**Timer 3 External Clock Select.** This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. Note: External clock sources are synchronized with the system clock. 00: External Clock is SYSCLK/12. Capture trigger is RTC. 01: External Clock is External Oscillator/8. Capture trigger is RTC. 10: External Clock is SYSCLK/12. Capture trigger is External Oscillator/8. 11: External Clock is RTC. Capture trigger is External Oscillator/8.

33.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 2. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) 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 2 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA counter high byte; the Module 2 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.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

33.4.1. 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 2 is forced into software timer mode.
- Writes to the Module 2 mode register (PCA0CPM2) 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 bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH2 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 PCA0CPH2. Upon a PCA0CPH2 write, PCA0H plus the offset held in PCA0CPL2 is loaded into PCA0CPH2 (See Figure 33.11).

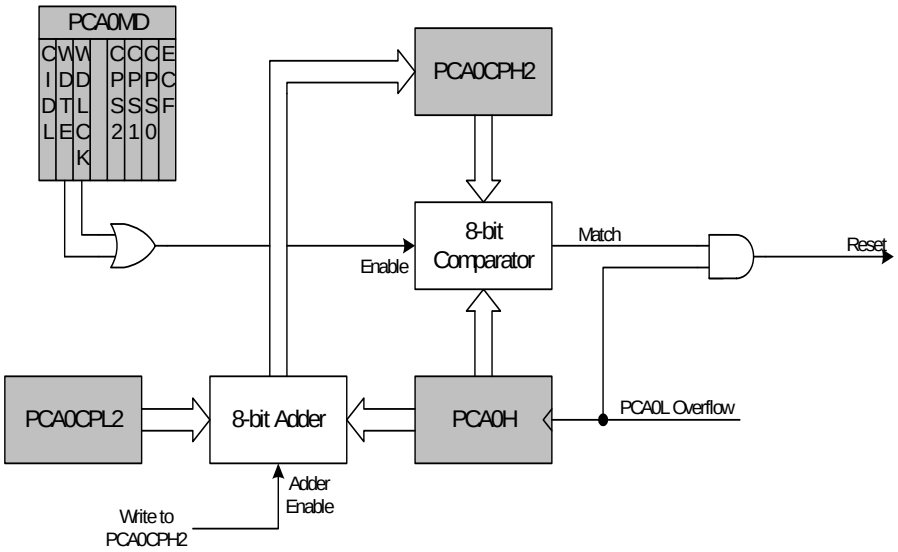

Figure 33.11. PCA Module 2 with Watchdog Timer Enabled

The 8-bit offset held in PCA0CPH2 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 33.5, where PCA0L is the value of the PCA0L register at the time of the update.

Register 33.12. PCA0CPM2: PCA Channel 2 Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDC								

Table 33.15. PCA0CPM2 Register Bit Descriptions

Bit	Name	Function
7	PWM16	**Channel 2 16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	**Channel 2 Comparator Function Enable.** This bit enables the comparator function.
5	CAPP	**Channel 2 Capture Positive Function Enable.** This bit enables the positive edge capture capability.
4	CAPN	**Channel 2 Capture Negative Function Enable.** This bit enables the negative edge capture capability.
3	MAT	**Channel 2 Match Function Enable.** This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF2 bit in the PCA0MD register to be set to logic 1.
2	TOG	**Channel 2 Toggle Function Enable.** This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX2 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	**Channel 2 Pulse Width Modulation Mode Enable.** This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX2 pin. 8 to 11-bit PWM is used if PWM16 is cleared to 0; 16-bit mode is used if PWM16 is set to 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	**Channel 2 Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCF2) interrupt. 0: Disable CCF2 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF2 is set.