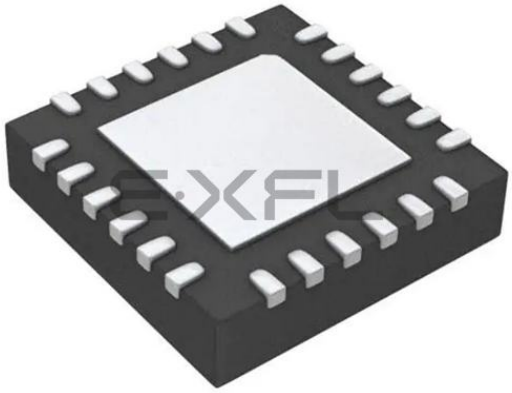

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	CIP-51 8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
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/c8051f987-gmr

Table 3.3. PCB Land Pattern

Dimension	Min	Max
D	2.71 REF	
D2	1.60	1.80
e	0.50 BSC	
E	2.71 REF	
E2	1.60	1.80
f	2.53 REF	
GD	2.10	—
GE	2.10	—
W	—	0.34
X	—	0.28
Y	0.61 REF	
ZE	—	3.31
ZD	—	3.31

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on IPC-SM-782 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

1. 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

1. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
2. The stencil thickness should be 0.125 mm (5 mils).
3. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
4. A 1.45 x 1.45 mm square aperture should be used for the center pad. This provides approximately 70% solder paste coverage on the pad, which is optimum to assure correct component stand-off.

Card Assembly

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

Table 3.5. 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

1. 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

1. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
2. The stencil thickness should be 0.125 mm (5 mils).
3. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
4. 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

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

SFR Definition 7.2. CPT0MD: Comparator 0 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP0RIE	CP0FIE			CP0MD[1:0]	
Type	R/W	R	R/W	R/W	R	R	R/W	
Reset	1	0	0	0	0	0	1	0

SFR Page = 0x0; SFR Address = 0x9D

Bit	Name	Function
7	Reserved	Read = 1b, Must Write 1b.
6	Unused	Read = 0b, Write = don't care.
5	CP0RIE	**Comparator0 Rising-Edge Interrupt Enable.** 0: Comparator0 Rising-edge interrupt disabled. 1: Comparator0 Rising-edge interrupt enabled.
4	CP0FIE	**Comparator0 Falling-Edge Interrupt Enable.** 0: Comparator0 Falling-edge interrupt disabled. 1: Comparator0 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP0MD[1:0]	**Comparator0 Mode Select** These bits affect the response time and power consumption for Comparator0. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

8.9. Automatic Scanning (Method 2—CS0SMEN = 1)

When CS0SMEN is enabled, CS0 uses an alternate autoscanning method that uses the contents of CS0SCAN0 and CS0SCAN1 to determine which channels to include in the scan. This maximizes flexibility for application development and can result in more power efficient scanning. The following procedure can be used to configure the device for Automatic Scanning with CS0SMEN = 1.

1. Set the CS0SMEN bit to 1.
2. Select the start of conversion mode (CS0CM[2:0]) if not already configured. Mode 101b is the mode of choice for most systems.
3. Configure the CS0SCAN0 and CS0SCAN1 registers to enable channels in the scan.
4. Configure the CS0THH:CS0THL digital comparator threshold and polarity.
5. Enable wake from suspend on end of scan (CS0WOI = 1) if this functionality is desired.
6. Set CS0SS to point to the first channel in the scan. Note: CS0SS uses the same bit mapping as the CS0MX register.
7. Issue a start of conversion (BUSY = 1).
8. Enable the CS0 Wakeup Source and place the device in Suspend mode (optional).

If using Mode 101b, scanning will stop once a “touch” has been detected using the digital comparator. The CS0MX register will contain the channel mux value of the channel that caused the interrupt. Setting the busy bit when servicing the interrupt will cause the scan to continue where it left off. Scanning will also stop after all channels have been sampled and no “touches” have been detected. If the CS0WOI bit is set, a wake from suspend event will be generated. Note: When automatic scanning is enabled, the contents of the CS0MX register are only valid when the digital comparator interrupt is set and BUSY = 0.

8.10. CS0 Comparator

The CS0 comparator compares the latest capacitive sense conversion result with the value stored in CS0THH:CS0THL. If the result is less than or equal to the stored value, the CS0CMPF bit(CS0CN:0) is set to 0. If the result is greater than the stored value, CS0CMPF is set to 1.

If the CS0 conversion accumulator is configured to accumulate multiple conversions, a comparison will not be made until the last conversion has been accumulated.

An interrupt will be generated if CS0 greater-than comparator interrupts are enabled by setting the ECSDC bit (EIE2.5) when the comparator sets CS0CMPF to 1.

If auto-scan is running when the comparator sets the CS0CMPF bit, no further auto-scan initiated conversions will start until firmware sets CS0BUSY to 1.

A CS0 greater-than comparator event can wake a device from suspend mode. This feature is useful in systems configured to continuously sample one or more capacitive sense channels. The device will remain in the low-power suspend state until the captured value of one of the scanned channels causes a CS0 greater-than comparator event to occur. It is not necessary to have CS0 comparator interrupts enabled in order to wake a device from suspend with a greater-than event.

For a summary of behavior with different CS0 comparator, auto-scan, and auto accumulator settings, please see Table 8.1.

SFR Definition 8.2. CS0CF: Capacitive Sense Configuration

Bit	7	6	5	4	3	2	1	0
Name	CS0SMEN	CS0CM[2:0]			CS0MCEN	CS0ACU[2:0]		
Type	R/W	R/W			R	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAA

Bit	Name	Description
7	CS0SMEN	**CS0 Channel Scan Masking Enable.** 0: The CS0SCAN0 and CS0SCAN1 register contents are ignored. 1: The CS0SCAN0 and CS0SCAN1 registers are used to determine which channels will be included in the scan.
6:4	CS0CM[2:0]	**CS0 Start of Conversion Mode Select.** 000: Conversion initiated on every write of 1 to CS0BUSY. 001: Conversion initiated on overflow of Timer 0. 010: Conversion initiated on overflow of Timer 2. 011: Conversion initiated on overflow of Timer 1. 100: Conversion initiated on overflow of Timer 3. **When CS0SMEN = 0** 101: Reserved. 110: Conversion initiated continuously on the channel selected by CS0MX after writing 1 to CS0BUSY. 111: Conversions initiated continuously on channels from CS0SS to CS0SE after writing 1 to CS0BUSY. **When CS0SMEN = 1** 101: Single Scan Mode, scans the channels selected by CS0SCAN0/1 once. 110: Conversion initiated continuously on the channel selected by CS0MX after writing 1 to CS0BUSY. 111: Auto Scan Mode, continuously scans the channels selected by CS0SCAN0/1.
3	CS0MCEN	**CS0 Multiple Channel Enable.** 0: Multiple channel feature is disabled. 1: Channels selected by CS0SCAN0/1 are internally shorted together and the combined node is selected as the CS0 input. This mode can be used to detect a capacitance change on multiple channels using a single conversion.
2:0	CS0ACU[2:0]	**CS0 Accumulator Mode Select.** 000: Accumulate 1 sample. 001: Accumulate 4 samples. 010: Accumulate 8 samples. 011: Accumulate 16 samples 100: Accumulate 32 samples. 101: Accumulate 64 samples. 11x: Reserved.

8.14. Capacitive Sense Multiplexer

The input multiplexer can be controlled through two methods. The CS0MX register can be written to through firmware, or the register can be configured automatically using the modules auto-scan functionality (see “8.8. Automatic Scanning (Method 1—CS0SMEN = 0)”).

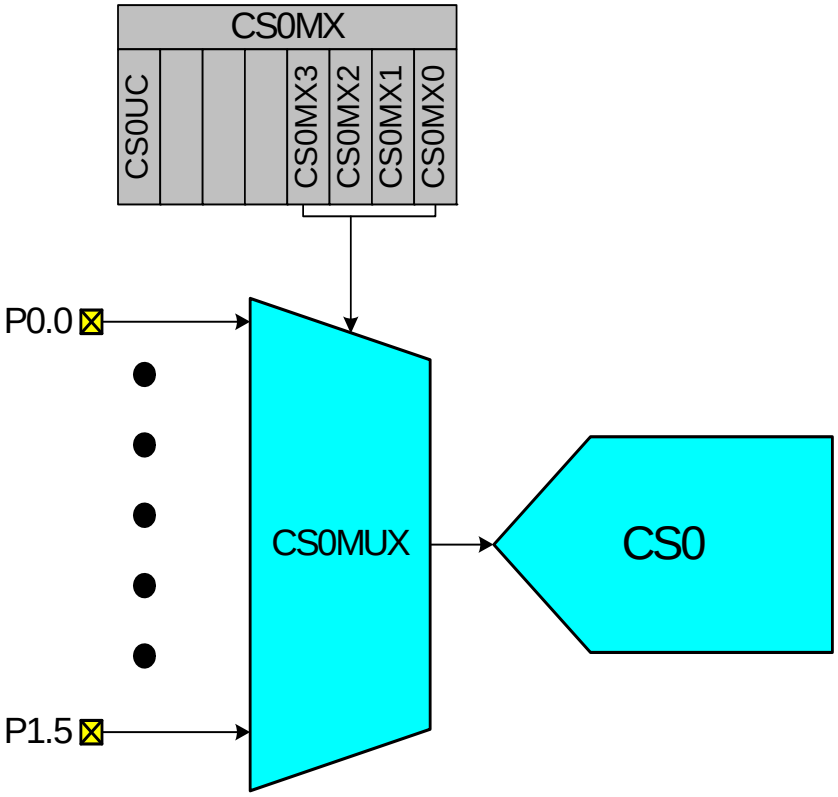

Figure 8.3. CS0 Multiplexer Block Diagram

Notes on Registers, Operands and Addressing Modes:

Rn—Register R0–R7 of the currently selected register bank.

@Ri—Data RAM location addressed indirectly through R0 or R1.

rel—8-bit, signed (twos complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).

#data—8-bit constant

#data16—16-bit constant

bit—Direct-accessed bit in Data RAM or SFR

addr11—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

addr16—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

SFR Definition 13.7. IT01CF: $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Page = 0x0; SFR Address = 0xE4

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7

SFR Definition 14.2. REVID: Revision Identification

Bit	7	6	5	4	3	2	1	0
Name	REVID[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xE2

Bit	Name	Function
7:0	REVID[7:0]	**Revision Identification.** These bits contain a value that can be decoded to determine the silicon revision. For example, 0x00 for Rev A, 0x01 for Rev B, 0x02 for Rev C, etc.

16.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRC0VAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRC0INIT).

16.3. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more 256 byte blocks. The following steps can be used to automatically perform a CRC on Flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRC0CNT.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute any additional code until the CRC operation completes. **See the note in SFR Definition 16.1. CRC0CN: CRC0 Control for more information on how to properly initiate a CRC calculation.**
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result using the procedure below.

16.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. The CRC0PNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

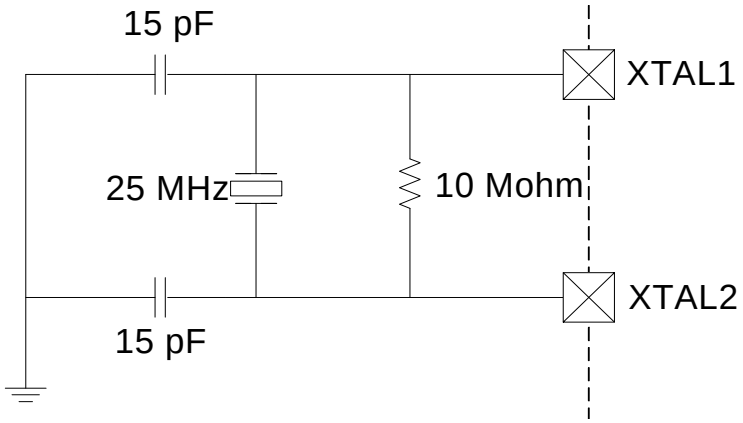

Figure 19.2. 25 MHz External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency. The selection should be based on Table 19.1. For example, a 25 MHz crystal requires an XFCN setting of 111b.

Table 19.1. Recommended XFCN Settings for Crystal Mode

XFCN	Crystal Frequency	Bias Current	Typical Supply Current (VDD = 2.4 V)
000	f ≤ 20 kHz	0.5 µA	3.0 µA, f = 32.768 kHz
001	20 kHz < f ≤ 58 kHz	1.5 µA	4.8 µA, f = 32.768 kHz
010	58 kHz < f ≤ 155 kHz	4.8 µA	9.6 µA, f = 32.768 kHz
011	155 kHz < f ≤ 415 kHz	14 µA	28 µA, f = 400 kHz
100	415 kHz < f ≤ 1.1 MHz	40 µA	71 µA, f = 400 kHz
101	1.1 MHz < f ≤ 3.1 MHz	120 µA	193 µA, f = 400 kHz
110	3.1 MHz < f ≤ 8.2 MHz	550 µA	940 µA, f = 8 MHz
111	8.2 MHz < f ≤ 25 MHz	2.6 mA	3.9 mA, f = 25 MHz

When the crystal oscillator is first enabled, the external oscillator valid detector allows software to determine when the external system clock has stabilized. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure for starting the crystal is as follows:

1. Configure XTAL1 and XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD $\geq$ 1.
4. Switch the system clock to the external oscillator.

20.3.3. Software Considerations for using the SmaRTClock Timer and Alarm

The SmaRTClock timer and alarm have two operating modes to suit varying applications. The two modes are described below:

Mode 1:
The first mode uses the SmaRTClock timer as a perpetual timebase which is never reset to zero. Every 36 hours, the timer is allowed to overflow without being stopped or disrupted. The alarm interval is software managed and is added to the ALRMn registers by software after each alarm. This allows the alarm match value to always stay ahead of the timer by one software managed interval. If software uses 32-bit unsigned addition to increment the alarm match value, then it does not need to handle overflows since both the timer and the alarm match value will overflow in the same manner.

This mode is ideal for applications which have a long alarm interval (e.g., 24 or 36 hours) and/or have a need for a perpetual timebase. An example of an application that needs a perpetual timebase is one whose wake-up interval is constantly changing. For these applications, software can keep track of the number of timer overflows in a 16-bit variable, extending the 32-bit (36 hour) timer to a 48-bit (272 year) perpetual timebase.

Mode 2:
The second mode uses the SmaRTClock timer as a general purpose up counter which is auto reset to zero by hardware after each alarm. The alarm interval is managed by hardware and stored in the ALRMn registers. Software only needs to set the alarm interval once during device initialization. After each alarm, software should keep a count of the number of alarms that have occurred in order to keep track of time.

This mode is ideal for applications that require minimal software intervention and/or have a fixed alarm interval. This mode is the most power efficient since it requires less CPU time per alarm.

21.1.3. Interfacing Port I/O to 5 V Logic

All Port I/O have internal ESD protection diodes to prevent the pin voltage from exceeding the V_{DD} supply. The Port I/O pins are not 5V tolerant and require level translators to interface to 5V logic.

21.1.4. Increasing Port I/O Drive Strength

Port I/O output drivers support a high and low drive strength; the default is low drive strength. The drive strength of a Port I/O can be configured using the PnDRV registers. See Section “4. Electrical Characteristics” on page 48 for the difference in output drive strength between the two modes.

21.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P1.7 can be assigned to various analog, digital, and external interrupt functions. The Port pins assuaged to analog functions should be configured for analog I/O and Port pins assuaged to digital or external interrupt functions should be configured for digital I/O.

21.2.1. Assigning Port I/O Pins to Analog Functions

Table 21.1 shows all available analog functions that need Port I/O assignments. **Port pins selected for these analog functions should have their digital drivers disabled (PnMDOUT.n = 0 and Port Latch = 1) and their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 21.1 shows the potential mapping of Port I/O to each analog function.

Table 21.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	Registers used for Assignment
ADC Input	P0.1–P0.7, P1.2–P1.4	ADC0MX, PnSKIP
Comparator0 Input	P1.0, P1.1	CPT0MX, PnSKIP
Voltage Reference (VREF0)	P0.0	REF0CN, PnSKIP
Analog Ground Reference (AGND)	P0.1	REF0CN, PnSKIP
Current Reference (IREF0)	P0.7	IREF0CN, PnSKIP
External Oscillator Input (XTAL1)	P0.2	OSCXCN, PnSKIP
External Oscillator Output (XTAL2)	P0.3	OSCXCN, PnSKIP
SmaRTClock Oscillator Input (XTAL3)	P1.6	RTC0CN, PnSKIP
SmaRTClock Oscillator Output (XTAL4)	P1.7	RTC0CN, PnSKIP

21.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 21.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 21.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SMBus, CP0 Outputs, System Clock Output, PCA0, Timer0 and Timer1 External Inputs.	Any Port pin available for assignment by the Crossbar. This includes P0.0–P1.7 pins which have their PnSKIP bit set to 0. **Note:** The Crossbar will always assign UART0 and SPI1 pins to fixed locations.	XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0–P1.7, P2.7	P0SKIP, P1SKIP

21.2.3. Assigning Port I/O Pins to External Digital Event Capture Functions

External digital event capture functions can be used to trigger an interrupt or wake the device from a low power mode when a transition occurs on a digital I/O pin. The digital event capture functions do not require dedicated pins and will function on both GPIO pins (PnSKIP = 1) and pins in use by the Crossbar (PnSKIP = 0). External digital even capture functions cannot be used on pins configured for analog I/O. Table 21.3 shows all available external digital event capture functions.

Table 21.3. Port I/O Assignment for External Digital Event Capture Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
External Interrupt 0	P0.0–P0.7	IT01CF
External Interrupt 1	P0.0–P0.7	IT01CF
Port Match	P0.0–P1.7	P0MASK, P0MAT P1MASK, P1MAT

22.4.4. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 22.5. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC2

Bit	Name	Function
7:0	SMB0DAT[7:0]	**SMBus Data.** The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

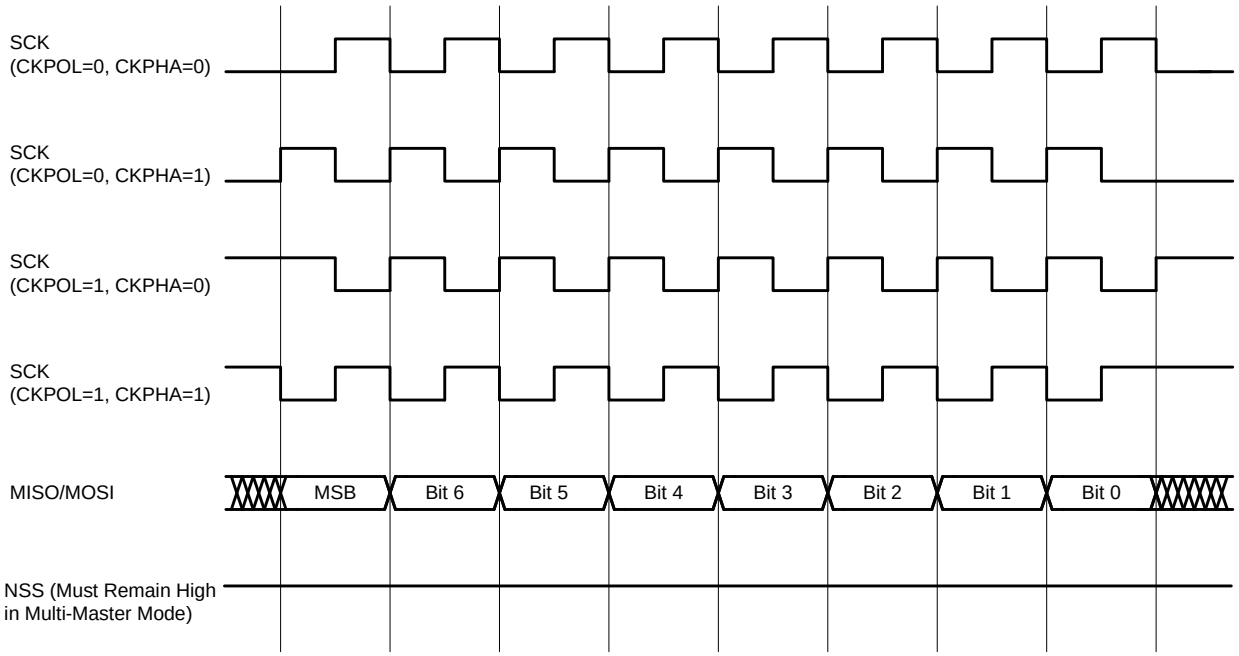

Figure 24.5. Master Mode Data/Clock Timing

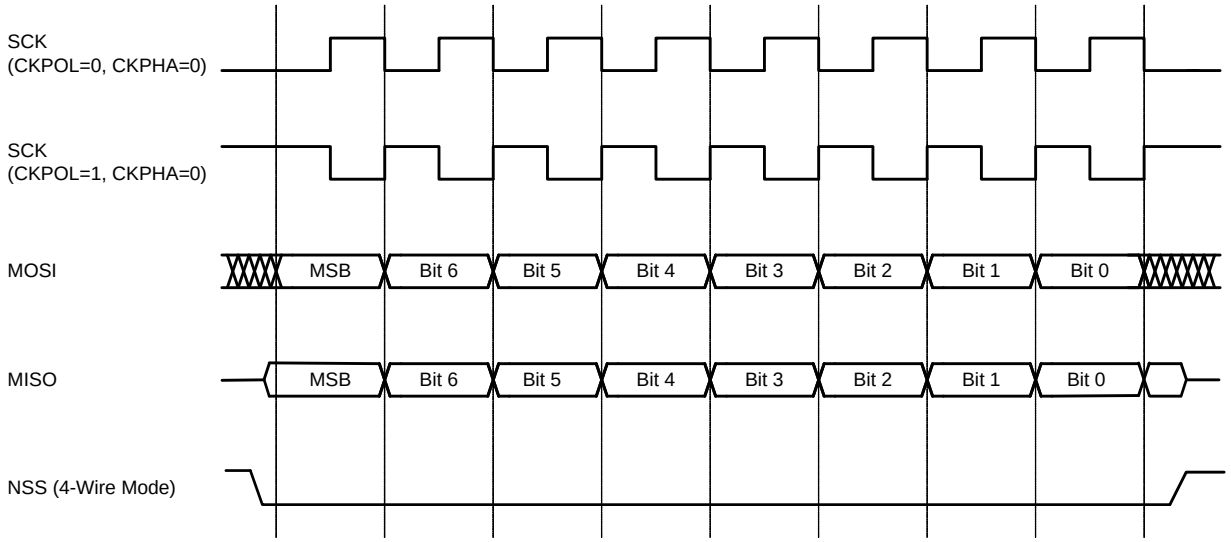

Figure 24.6. Slave Mode Data/Clock Timing (CKPHA = 0)

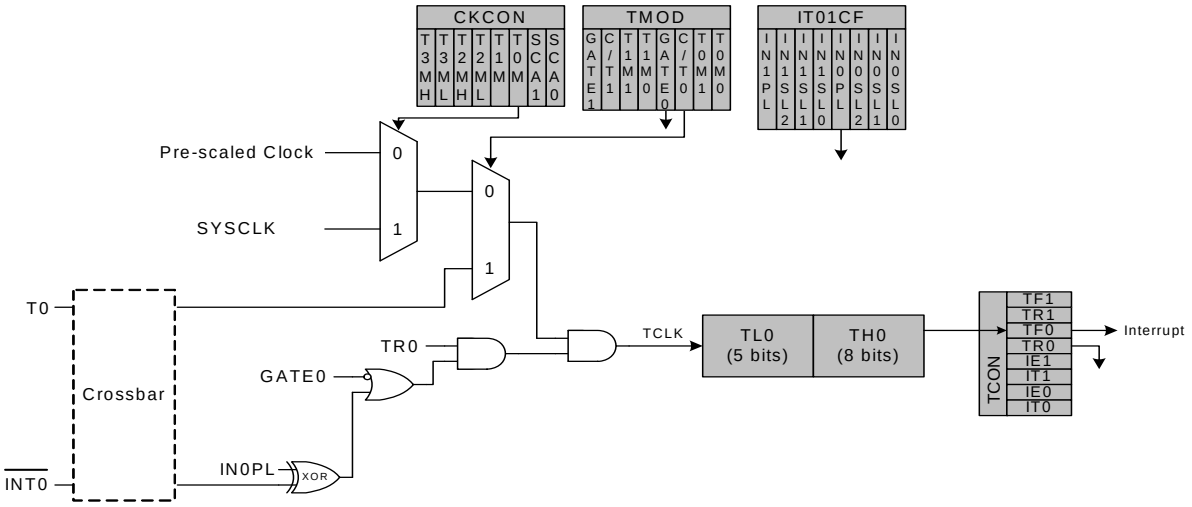

Figure 25.1. T0 Mode 0 Block Diagram

25.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

25.3.2. 8-Bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 25.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, the external oscillator clock source divided by 8, or the SmaRTClock. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bits (T3XCLK[1:0] in TMR3CN), as follows:

T3MH	T3XCLK[1:0]	TMR3H Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK[1:0]	TMR3L Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

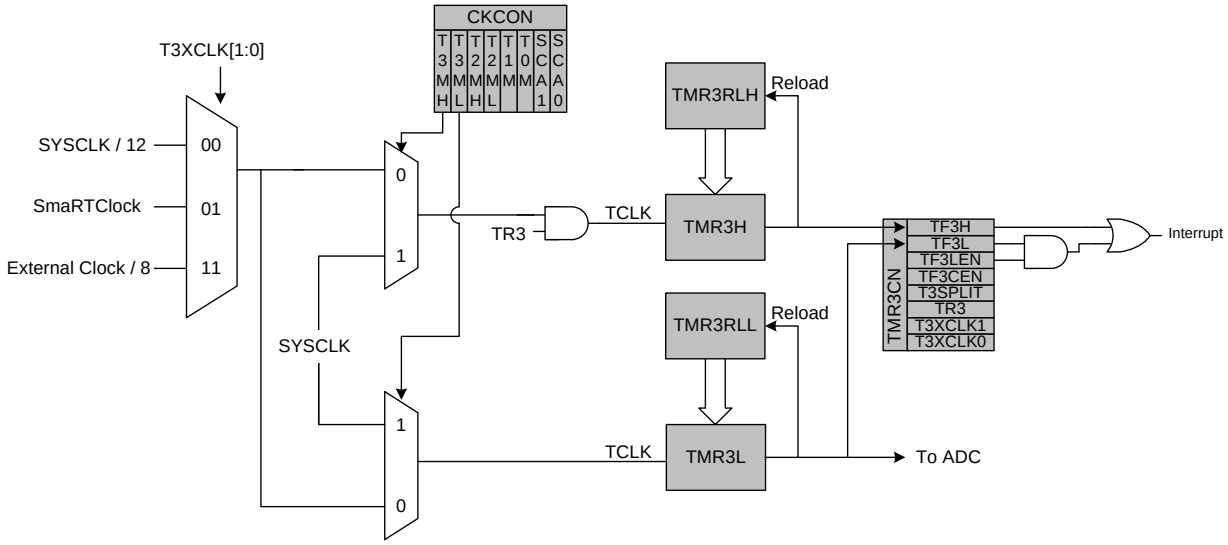

Figure 25.8. Timer 3 8-Bit Mode Block Diagram

SFR Definition 26.5. PCA0L: PCA Counter/Timer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xF9

Bit	Name	Function
7:0	PCA0[7:0]	**PCA Counter/Timer Low Byte.** The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

Note: When the WDTE bit is set to 1, the PCA0L register cannot be modified by software. To change the contents of the PCA0L register, the Watchdog Timer must first be disabled.

SFR Definition 26.6. PCA0H: PCA Counter/Timer High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xFA

Bit	Name	Function
7:0	PCA0[15:8]	**PCA Counter/Timer High Byte.** The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Reads of this register will read the contents of a “snapshot” register, whose contents are updated only when the contents of PCA0L are read (see Section 26.1).

Note: When the WDTE bit is set to 1, the PCA0H register cannot be modified by software. To change the contents of the PCA0H register, the Watchdog Timer must first be disabled.