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	Discontinued at Digi-Key
Core Processor	8051
Core Size	8-Bit
Speed	48 MIPS
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.25V
Data Converters	A/D 32x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f388-gq

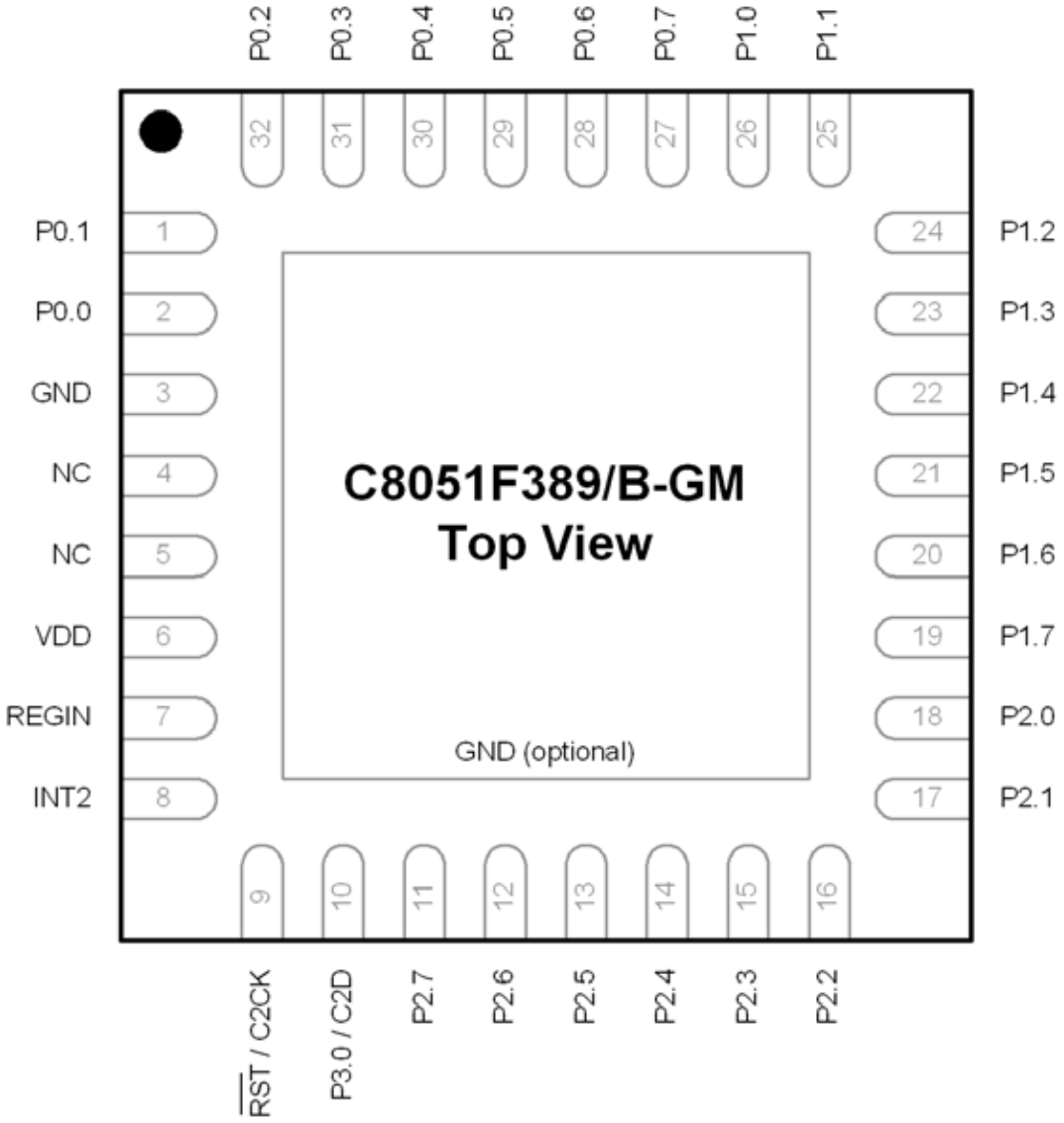

Figure 3.7. QFN-32 (C8051F389/B) Pinout Diagram (Top View)

6. 10-Bit ADC

ADC0 on the C8051F388/9/A/B is a 500 ksps, 10-bit successive-approximation-register (SAR) ADC with integrated track-and-hold, and a programmable window detector. The ADC is fully configurable under software control via Special Function Registers. The ADC may be configured to measure various different signals using the analog multiplexer described in Section “6.5. ADC0 Analog Multiplexer” on page 55. The voltage reference for the ADC is selected as described in Section “7. Voltage Reference Options” on page 58. The ADC0 subsystem is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

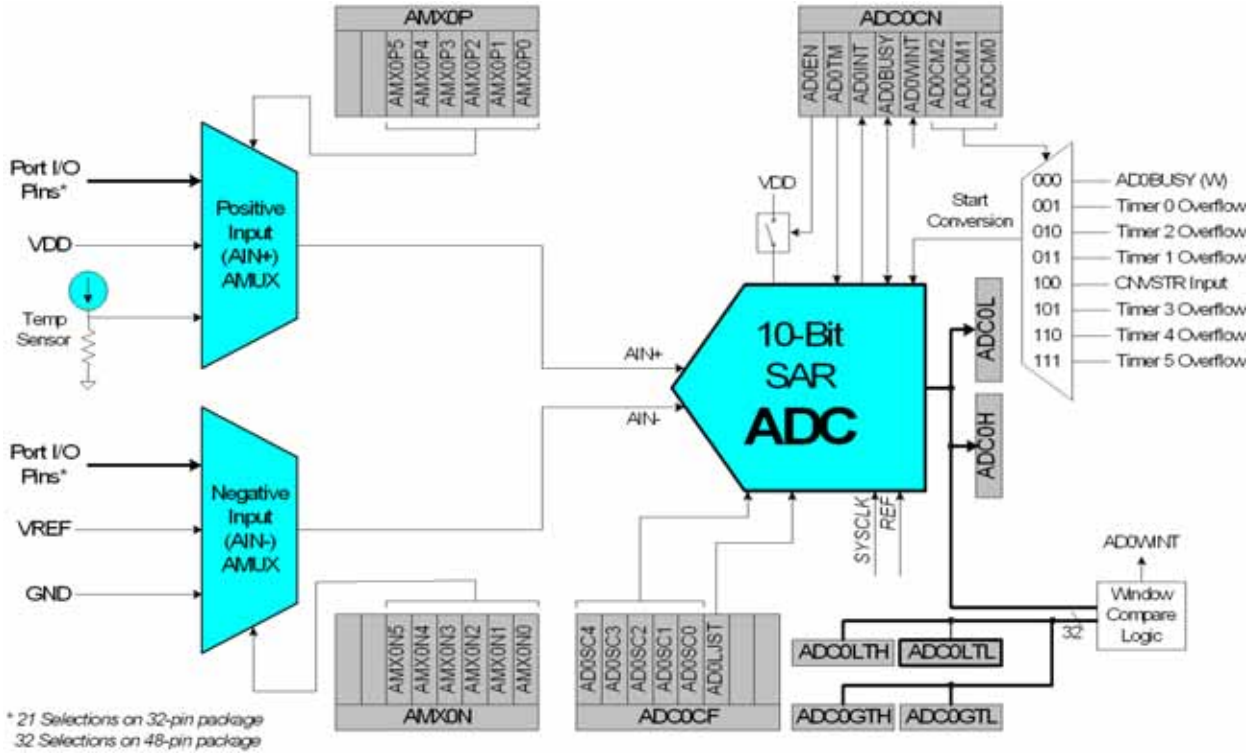

Figure 6.1. ADC0 Functional Block Diagram

6.2. Temperature Sensor

The typical temperature sensor transfer function is shown in Figure 6.2. The output voltage (V_{TEMP}) is the positive ADC input when the temperature sensor is selected by bits AMX0P5-0 in register AMX0P.

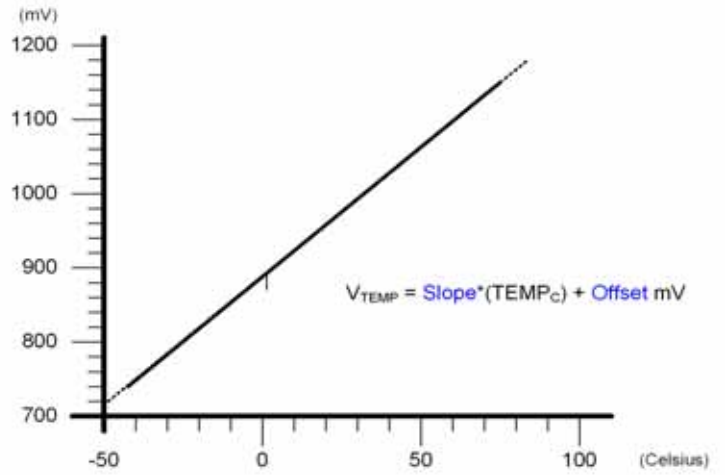

Figure 6.2. Typical Temperature Sensor Transfer Function

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.10, “ADC0 Electrical Characteristics,” on page 39 for linearity specifications). For absolute temperature measurements, gain and/or offset calibration is recommended. Typically a 1-point calibration includes the following steps:

Step 1. Control/measure the ambient temperature (this temperature must be known).
Step 2. Power the device, and delay for a few seconds to allow for self-heating.
Step 3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
Step 4. Calculate the offset and/or gain characteristics, and store these values in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 6.3 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Note that parameters which affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

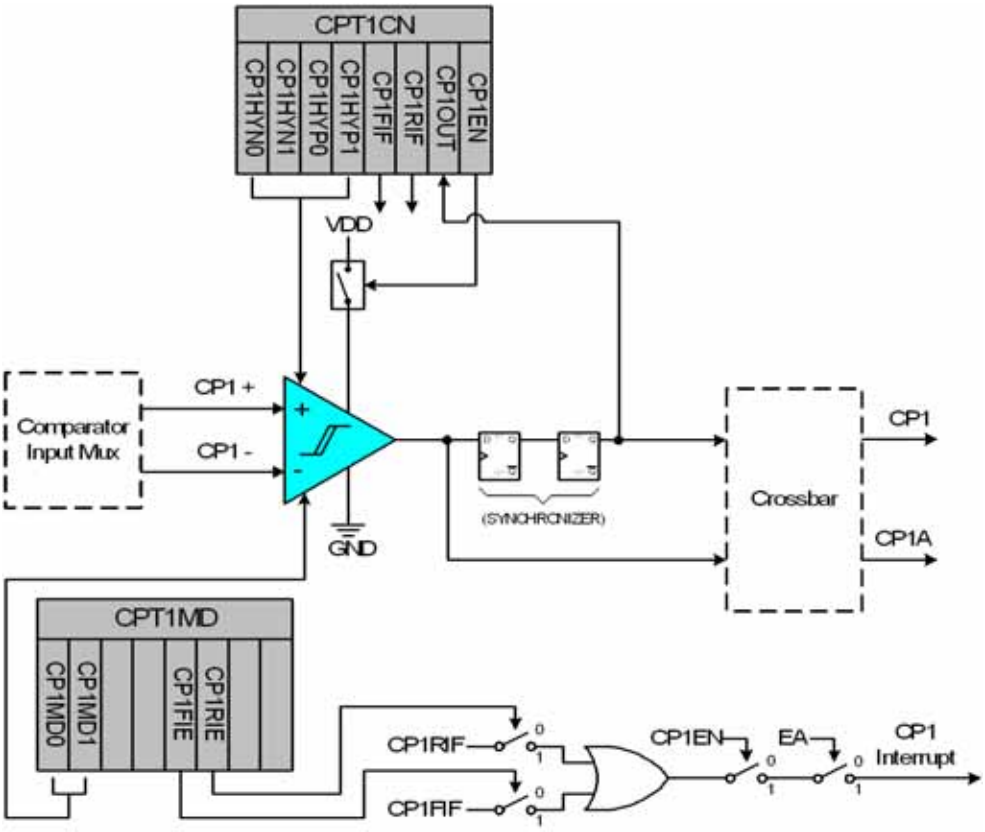

Figure 8.2. Comparator1 Functional Block Diagram

The Comparator output can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and the power supply to the comparator is turned off. See Section “20.1. Priority Crossbar Decoder” on page 148 for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from –0.25 V to (V_{DD}) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Section “5. Electrical Characteristics” on page 34.

The Comparator response time may be configured in software via the CPTnMD registers (see SFR Definition 8.2 and SFR Definition 8.4). Selecting a longer response time reduces the Comparator supply current.

Table 11.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2

Table 15.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Page	Description	Page
TMR5RLL	0xCA	F	Timer/Counter 5 Reload Low	257
VDM0CN	0xFF	All Pages	V_{DD} Monitor Control	126
XBR0	0xE1	All Pages	Port I/O Crossbar Control 0	153
XBR1	0xE2	All Pages	Port I/O Crossbar Control 1	154
XBR2	0xE3	All Pages	Port I/O Crossbar Control 2	155

SFR Definition 16.2. IP: Interrupt Priority

Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0xB8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1b, Write = Don't Care.
6	PSPI0	**Serial Peripheral Interface (SPI0) Interrupt Priority Control.** This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	**Timer 2 Interrupt Priority Control.** This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	**UART0 Interrupt Priority Control.** This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	**Timer 1 Interrupt Priority Control.** This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	**External Interrupt 1 Priority Control.** This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	**Timer 0 Interrupt Priority Control.** This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	**External Interrupt 0 Priority Control.** This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

19.2. Programmable Internal High-Frequency (H-F) Oscillator

All C8051F388/9/A/B devices include a programmable internal high-frequency oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register as defined by SFR Definition 19.2.

On C8051F388/9/A/B devices, OSCICL is factory calibrated to obtain a 48 MHz base frequency. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8 after a divide by 4 stage, as defined by the IFCN bits in register OSCICN. The divide value defaults to 8 following a reset, which results in a 1.5 MHz system clock.

19.2.1. Internal Oscillator Suspend Mode

When software writes a logic 1 to SUSPEND (OSCICN.5), the internal oscillator is suspended. If the system clock is derived from the internal oscillator, the input clock to the peripheral or CIP-51 will be stopped until a rising or falling edge occurs on the INT2 pin.

When one of the oscillator awakening events occur, the internal oscillator, CIP-51, and affected peripherals resume normal operation. The CPU resumes execution at the instruction following the write to the SUSPEND bit.

Note: The prefetch engine can be turned off in suspend mode to save power. Additionally, both Voltage Regulators (REG0 and REG1) have low-power modes for additional power savings in suspend mode.

SFR Definition 19.2. OSCICL: Internal H-F Oscillator Calibration

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

SFR Address = 0xB3; SFR Page = All Pages

Bit	Name	Function
7	Unused	Read = 0; Write = don't care
6:0	OSCICL[6:0]	**Internal Oscillator Calibration Bits.** These bits determine the internal oscillator period. When set to 0000000b, the H-F oscillator operates at its fastest setting. When set to 1111111b, the H-F oscillator operates at its slowest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 48 MHz. OSCICL should only be changed by firmware when the H-F oscillator is disabled (IOSCEN = 0).

SFR Definition 19.3. OSCICN: Internal H-F Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY	SUSPEND				IFCN[1:0]	
Type	R/W	R	R/W	R	R	R	R/W	
Reset	1	1	0	0	0	0	0	0

SFR Address = 0xB2; SFR Page = All Pages

Bit	Name	Function
7	IOSCEN	**Internal H-F Oscillator Enable Bit.** 0: Internal H-F Oscillator Disabled. 1: Internal H-F Oscillator Enabled.
6	IFRDY	**Internal H-F Oscillator Frequency Ready Flag.** 0: Internal H-F Oscillator is not running at programmed frequency. 1: Internal H-F Oscillator is running at programmed frequency.
5	SUSPEND	**Internal Oscillator Suspend Enable Bit.** Setting this bit to logic 1 places the internal oscillator in SUSPEND mode. The internal oscillator resumes operation when one of the SUSPEND mode awakening events occurs.
4:2	Unused	Read = 000b; Write = don't care
1:0	IFCN[1:0]	**Internal H-F Oscillator Frequency Divider Control Bits.** The Internal H-F Oscillator is divided by the IFCN bit setting after a divide-by-4 stage. 00: SYSCLK can be derived from Internal H-F Oscillator divided by 8 (1.5 MHz). 01: SYSCLK can be derived from Internal H-F Oscillator divided by 4 (3 MHz). 10: SYSCLK can be derived from Internal H-F Oscillator divided by 2 (6 MHz). 11: SYSCLK can be derived from Internal H-F Oscillator divided by 1 (12 MHz).

SFR Definition 20.18. P3MDOUT: Port 3 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA7; SFR Page = All Pages

Bit	Name	Function
7:0	P3MDOUT[7:0]	**Output Configuration Bits for P3.7–P3.0 (respectively).** These bits are ignored if the corresponding bit in register P3MDIN is logic 0. 0: Corresponding P3.n Output is open-drain. 1: Corresponding P3.n Output is push-pull.

SFR Definition 20.19. P3SKIP: Port 3 Skip

Bit	7	6	5	4	3	2	1	0
Name	P3SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDF; SFR Page = All Pages

Bit	Name	Function
7:0	P3SKIP[3:0]	**Port 3 Crossbar Skip Enable Bits.** These bits select Port 3 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P3.n pin is not skipped by the Crossbar. 1: Corresponding P3.n pin is skipped by the Crossbar.

SFR Definition 20.22. P4MDOUT: Port 4 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P4MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAE; SFR Page = All Pages

Bit	Name	Function
7:0	P4MDOUT[7:0]	**Output Configuration Bits for P4.7–P4.0 (respectively).** These bits are ignored if the corresponding bit in register P4MDIN is logic 0. 0: Corresponding P4.n Output is open-drain. 1: Corresponding P4.n Output is push-pull.

21.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. The position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur **after** the ACK, regardless of whether hardware ACK generation is enabled or not.

21.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. The interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 21.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

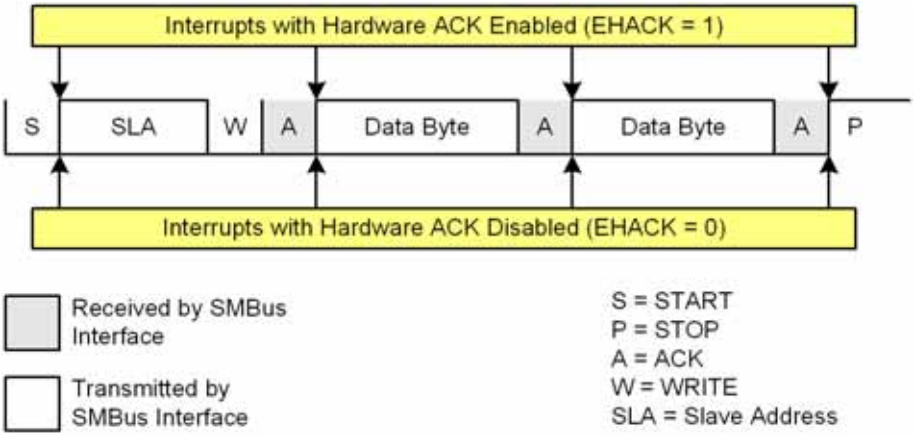

Figure 21.5. Typical Master Write Sequence

SFR Definition 25.2. CKCON1: Clock Control 1

Bit	7	6	5	4	3	2	1	0
Name					T5MH	T5ML	T4MH	T4ML
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE4; SFR Page = F

Bit	Name	Function
7:4	Unused	Read = 0000b; Write = don't care
3	T5MH	**Timer 5 High Byte Clock Select.** Selects the clock supplied to the Timer 5 high byte (split 8-bit timer mode only). 0: Timer 5 high byte uses the clock defined by the T5XCLK bit in TMR5CN. 1: Timer 5 high byte uses the system clock.
2	T5ML	**Timer 5 Low Byte Clock Select.** Selects the clock supplied to Timer 5. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 5 low byte uses the clock defined by the T5XCLK bit in TMR5CN. 1: Timer 5 low byte uses the system clock.
1	T4MH	**Timer 4 High Byte Clock Select.** Selects the clock supplied to the Timer 4 high byte (split 8-bit timer mode only). 0: Timer 4 high byte uses the clock defined by the T4XCLK bit in TMR4CN. 1: Timer 4 high byte uses the system clock.
0	T4ML	**Timer 4 Low Byte Clock Select.** Selects the clock supplied to Timer 4. If Timer 4 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 4 low byte uses the clock defined by the T4XCLK bit in TMR4CN. 1: Timer 4 low byte uses the system clock.

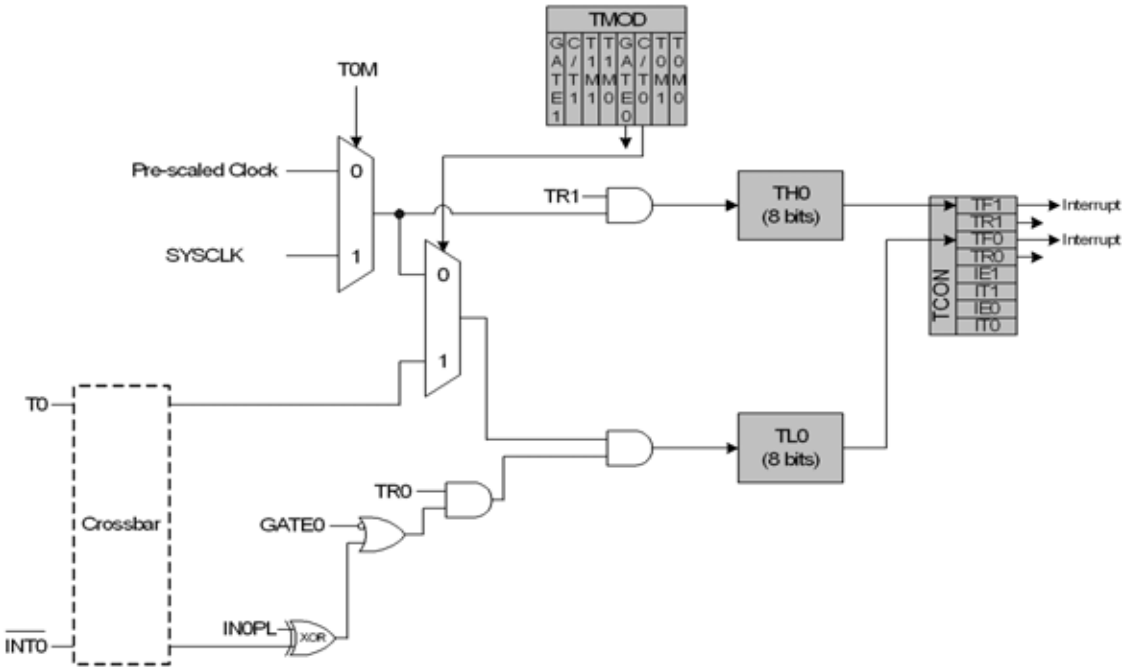

Figure 25.3. T0 Mode 3 Block Diagram

SFR Definition 25.5. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8A; SFR Page = All Pages

Bit	Name	Function
7:0	TL0[7:0]	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 25.6. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8B; SFR Page = All Pages

Bit	Name	Function
7:0	TL1[7:0]	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

25.2. 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, (split) 8-bit auto-reload mode, or Low-Frequency Oscillator (LFO) Falling Edge capture mode. The Timer 2 operation mode is defined by the T2SPLIT (TMR2CN.3), T2CE (TMR2CN.4) bits, and T2CSS (TMR2CN.1) bits.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

25.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. 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 25.4, 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 TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

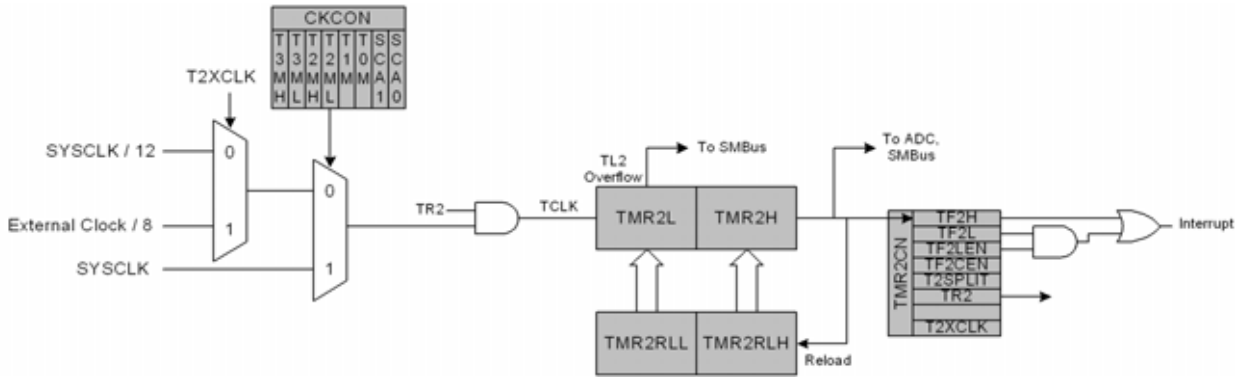

Figure 25.4. Timer 2 16-Bit Mode Block Diagram

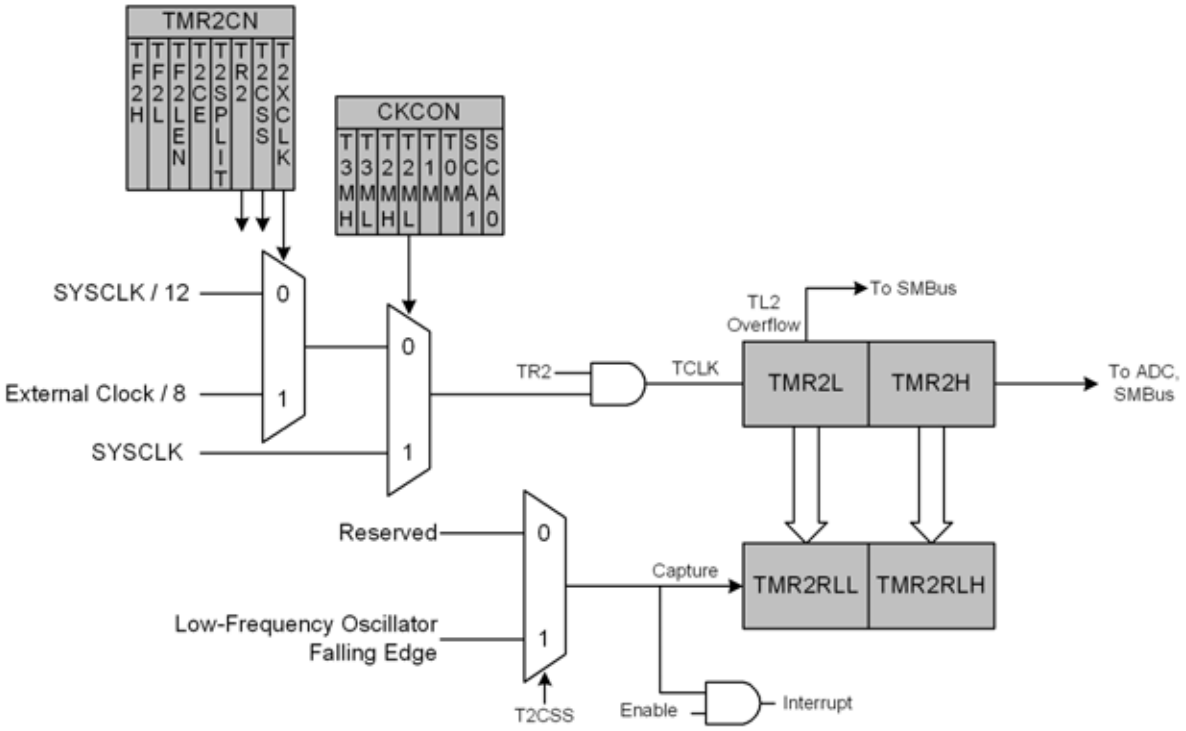

Figure 25.6. Timer 2 Capture Mode (T2SPLIT = 0)

When T2SPLIT = 1, the Timer 2 registers (TMR2H and TMR2L) act as two 8-bit counters. Each counter counts up independently and overflows from 0xFF to 0x00. Each time a capture event is received, the contents of the Timer 2 registers are latched into the Timer 2 Reload registers (TMR2RLH and TMR2RLL). A Timer 2 interrupt is generated if enabled.

SFR Definition 26.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK		CPS[2:0]			ECF
Type	R/W	R/W	R/W	R	R/W			R/W
Reset	0	1	0	0	0	0	0	0

SFR Address = 0xD9; SFR Page = All Pages

Bit	Name	Function
7	CIDL	**PCA Counter/Timer Idle Control.** Specifies PCA behavior when CPU is in Idle Mode. 0: PCA continues to function normally while the system controller is in Idle Mode. 1: PCA operation is suspended while the system controller is in Idle Mode.
6	WDTE	**Watchdog Timer Enable.** If this bit is set, PCA Module 4 is used as the watchdog timer. 0: Watchdog Timer disabled. 1: PCA Module 4 enabled as Watchdog Timer.
5	WDLCK	**Watchdog Timer Lock.** This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Unused	Read = 0b, Write = Don't care.
3:1	CPS[2:0]	**PCA Counter/Timer Pulse Select.** These bits select the timebase source for the PCA counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 11x: Reserved
0	ECF	**PCA Counter/Timer Overflow Interrupt Enable.** This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt. 0: Disable the CF interrupt. 1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.
Note: When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.		

SFR Definition 26.4. 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 Address = 0xF9; SFR Page = All Pages

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 26.5. 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 Address = 0xFA; SFR Page = All Pages

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.