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	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	40
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f348-gqr

11.2.Power-Fail Reset / VDD Monitor 102
11.3.External Reset 103
11.4.Missing Clock Detector Reset 103
11.5.Comparator0 Reset 103
11.6.PCA Watchdog Timer Reset 103
11.7.Flash Error Reset 103
11.8.Software Reset 104
11.9.USB Reset 104
12. Flash Memory 107
12.1.Programming The Flash Memory 107
12.1.1.Flash Lock and Key Functions 107
12.1.2.Flash Erase Procedure 107
12.1.3.Flash Write Procedure 108
12.2.Non-Volatile Data Storage 109
12.3.Security Options 109
13. External Data Memory Interface and On-Chip XRAM 114
13.1.Accessing XRAM 114
13.1.1.16-Bit MOVX Example 114
13.1.2.8-Bit MOVX Example 114
13.2.Accessing USB FIFO Space 115
13.3.Configuring the External Memory Interface 116
13.4.Port Configuration 116
13.5.Multiplexed and Non-multiplexed Selection 119
13.5.1.Multiplexed Configuration 119
13.5.2.Non-multiplexed Configuration 120
13.6.Memory Mode Selection 120
13.6.1.Internal XRAM Only 121
13.6.2.Split Mode without Bank Select 121
13.6.3.Split Mode with Bank Select 122
13.6.4.External Only 122
13.7.Timing 122
13.7.1.Non-multiplexed Mode 124
13.7.2.Multiplexed Mode 127
14. Oscillators 131
14.1.Programmable Internal High-Frequency (H-F) Oscillator 132
14.1.1.Internal H-F Oscillator Suspend Mode 132
14.2.Programmable Internal Low-Frequency (L-F) Oscillator 133
14.2.1.Calibrating the Internal L-F Oscillator 133
14.3.External Oscillator Drive Circuit 135
14.3.1.Clocking Timers Directly Through the External Oscillator 135
14.3.2.External Crystal Example 135
14.3.3.External RC Example 136
14.3.4.External Capacitor Example 136
14.4.4x Clock Multiplier 138
14.5.System and USB Clock Selection 139

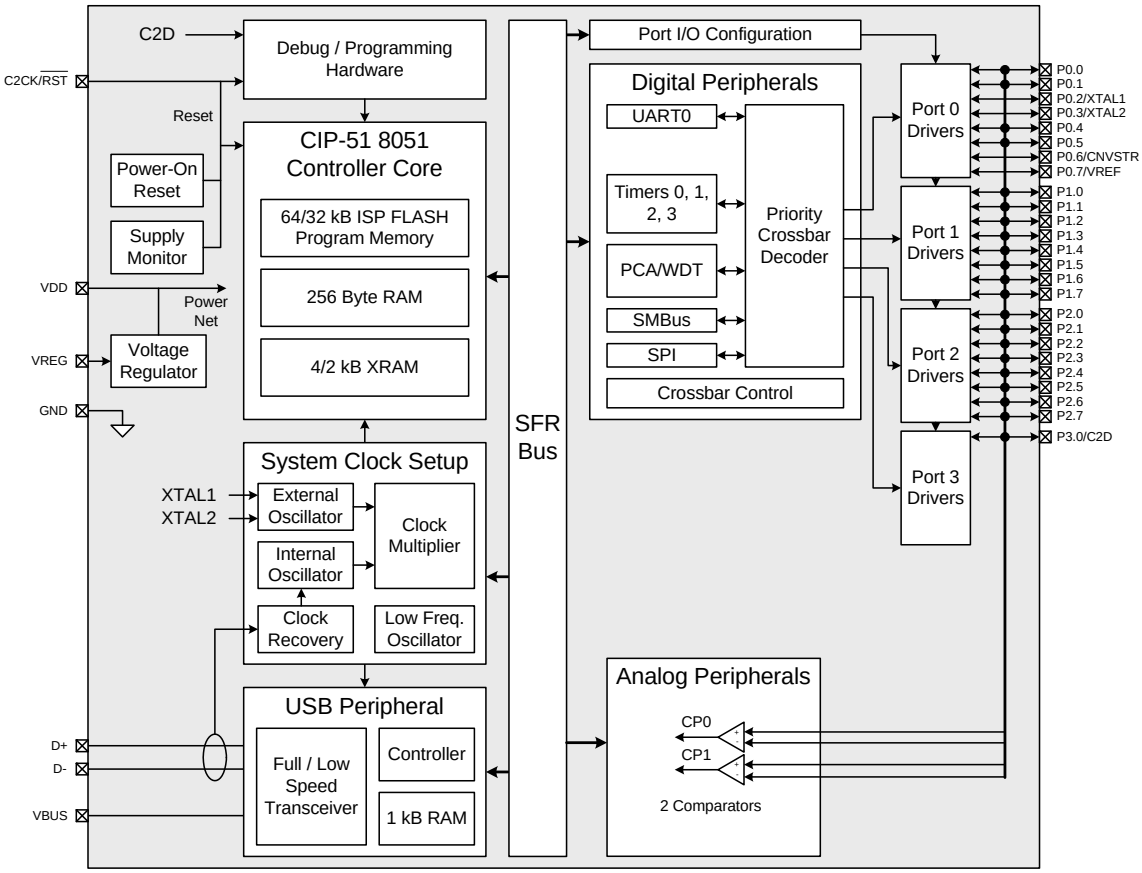

Figure 1.4. C8051F349/D Block Diagram

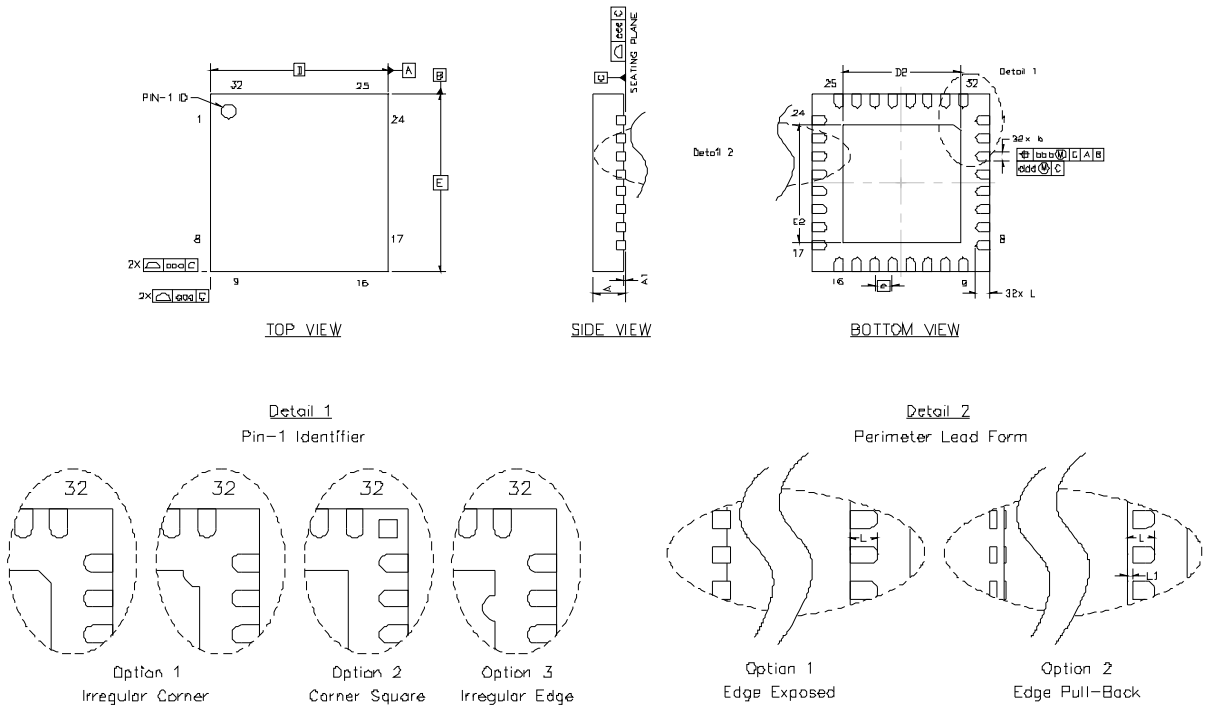

Figure 4.8. QFN-32 Package Drawing

Table 4.6. QFN-32 Package Dimensions

Dimension	Min	Nom	Max
A	0.80	0.9	1.00
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D		5.00 BSC	
D2	3.20	3.30	3.40
e		0.50 BSC	
E		5.00 BSC	
E2	3.20	3.30	3.40
L	0.30	0.40	0.50

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC Solid State Outline MO-220, variation VHHD except for custom features D2, E2, and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

7. Comparators

C8051F34x devices include two on-chip programmable voltage Comparators. A block diagram of the comparators is shown in Figure 7.1, where “n” is the comparator number (0 or 1). The two Comparators operate identically with the following exceptions: (1) Their input selections differ, and (2) Comparator0 can be used as a reset source. For input selection details, refer to SFR Definition 7.2 and SFR Definition 7.5.

Each Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous signal is available even when the system clock is not active. This allows the Comparators to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator outputs may be configured as open drain or push-pull (see **Section “15.2. Port I/O Initialization” on page 147**). Comparator0 may also be used as a reset source (see **Section “11.5. Comparator0 Reset” on page 103**).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input. The Comparator1 inputs are selected in the CPT1MX register (SFR Definition 7.5). The CMX1P1-CMX1P0 bits select the Comparator1 positive input; the CMX1N1-CMX1N0 bits select the Comparator1 negative input.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see **Section “15.3. General Purpose Port I/O” on page 150**).

SFR Definition 7.6. CPT1MD: Comparator1 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP1RIE	CP1FIE	-	-	CP1MD1	CP1MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9C

Bits7–6: UNUSED. Read = 00b, Write = don't care.
Bit5: CP1RIE: Comparator1 Rising-Edge Interrupt Enable.
0: Comparator1 rising-edge interrupt disabled.
1: Comparator1 rising-edge interrupt enabled.
Bit4: CP1FIE: Comparator1 Falling-Edge Interrupt Enable.
0: Comparator1 falling-edge interrupt disabled.
1: Comparator1 falling-edge interrupt enabled.
Bits1–0: CP1MD1–CP1MD0: Comparator1 Mode Select.
These bits select the response time for Comparator1.

Mode	CP1MD1	CP1MD0	CP1 Response Time*
0	0	0	Fastest Response
1	0	1	
2	1	0	
3	1	1	Lowest Power

* See Table 7.1 for response time parameters.

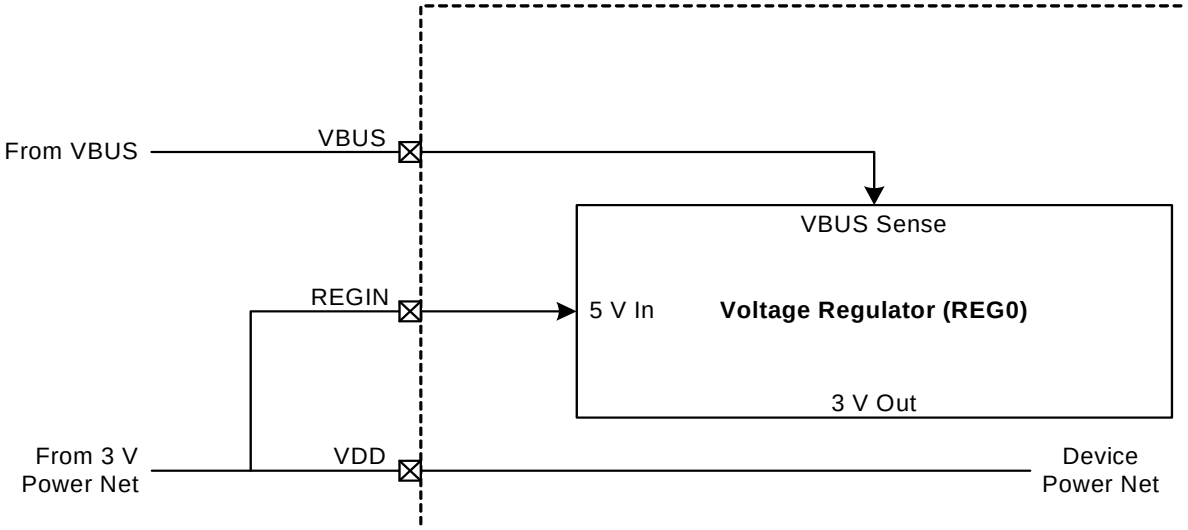

Figure 8.3. REG0 Configuration: USB Self-Powered, Regulator Disabled

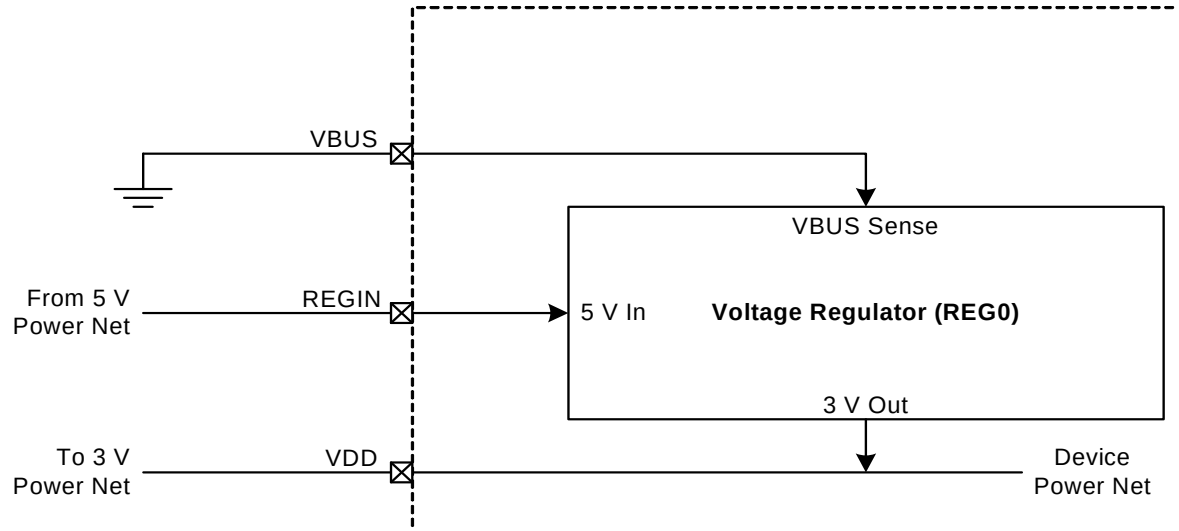

Figure 8.4. REG0 Configuration: No USB Connection

9. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. Included are four 16-bit counter/timers (see description in **Section 21**), an enhanced full-duplex UART (see description in **Section 18**), an Enhanced SPI (see description in **Section 20**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (**Section 9.2.6**), and 25 Port I/O (see description in **Section 15**). The CIP-51 also includes on-chip debug hardware (see description in **Section 23**), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 9.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 0 to 48 MHz Clock Frequency
- 256 Bytes of Internal RAM
- 25 Port I/O
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

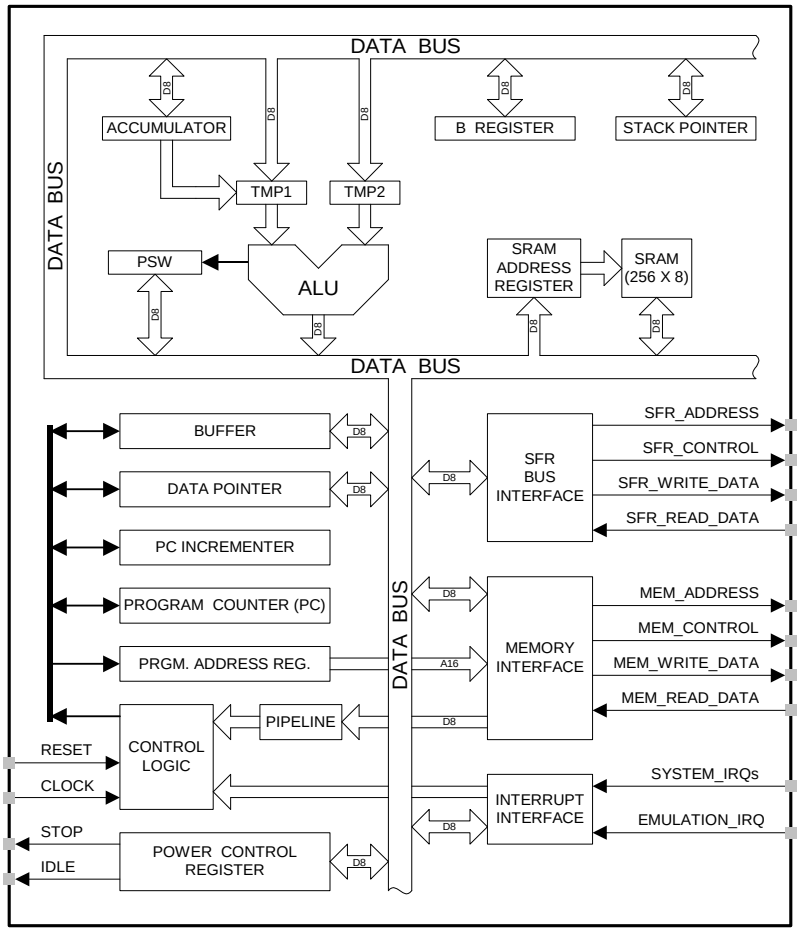

Figure 9.1. CIP-51 Block Diagram

11.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pull-up and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 11.1 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

11.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If more than 100 µs pass between rising edges on the system clock, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read ‘1’, signifying the MCD as the reset source; otherwise, this bit reads ‘0’. Writing a ‘1’ to the MCDRSF bit enables the Missing Clock Detector; writing a ‘0’ disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

11.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a ‘1’ to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), a system reset is generated. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read ‘1’ signifying Comparator0 as the reset source; otherwise, this bit reads ‘0’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

11.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in **Section “22.3. Watchdog Timer Mode” on page 264**; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to ‘1’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

11.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to “1”, and a MOVX write operation is attempted above address 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Flash read is attempted above user code space. This occurs when a MOVC operation is attempted above address 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see **Section “12.3. Security Options” on page 109**).
- A Flash Write or Erase is attempted when the V_{DD} monitor is not enabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

SFR Definition 13.3. EMI0TC: External Memory Timing Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EAS1	EAS0	EWR3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x84

Bits7–6: EAS1–0: EMIF Address Setup Time Bits.
00: Address setup time = 0 SYSCLK cycles.
01: Address setup time = 1 SYSCLK cycle.
10: Address setup time = 2 SYSCLK cycles.
11: Address setup time = 3 SYSCLK cycles.

Bits5–2: EWR3–0: EMIF $\overline{WR}$ and $\overline{RD}$ Pulse-Width Control Bits.
0000: $\overline{WR}$ and $\overline{RD}$ pulse width = 1 SYSCLK cycle.
0001: $\overline{WR}$ and $\overline{RD}$ pulse width = 2 SYSCLK cycles.
0010: $\overline{WR}$ and $\overline{RD}$ pulse width = 3 SYSCLK cycles.
0011: $\overline{WR}$ and $\overline{RD}$ pulse width = 4 SYSCLK cycles.
0100: $\overline{WR}$ and $\overline{RD}$ pulse width = 5 SYSCLK cycles.
0101: $\overline{WR}$ and $\overline{RD}$ pulse width = 6 SYSCLK cycles.
0110: $\overline{WR}$ and $\overline{RD}$ pulse width = 7 SYSCLK cycles.
0111: $\overline{WR}$ and $\overline{RD}$ pulse width = 8 SYSCLK cycles.
1000: $\overline{WR}$ and $\overline{RD}$ pulse width = 9 SYSCLK cycles.
1001: $\overline{WR}$ and $\overline{RD}$ pulse width = 10 SYSCLK cycles.
1010: $\overline{WR}$ and $\overline{RD}$ pulse width = 11 SYSCLK cycles.
1011: $\overline{WR}$ and $\overline{RD}$ pulse width = 12 SYSCLK cycles.
1100: $\overline{WR}$ and $\overline{RD}$ pulse width = 13 SYSCLK cycles.
1101: $\overline{WR}$ and $\overline{RD}$ pulse width = 14 SYSCLK cycles.
1110: $\overline{WR}$ and $\overline{RD}$ pulse width = 15 SYSCLK cycles.
1111:$\overline{WR}$ and $\overline{RD}$ pulse width = 16 SYSCLK cycles.

Bits1–0: EAH1–0: EMIF Address Hold Time Bits.
00: Address hold time = 0 SYSCLK cycles.
01: Address hold time = 1 SYSCLK cycle.
10: Address hold time = 2 SYSCLK cycles.
11: Address hold time = 3 SYSCLK cycles.

13.7.1. Non-multiplexed Mode

13.7.1.1.16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’.

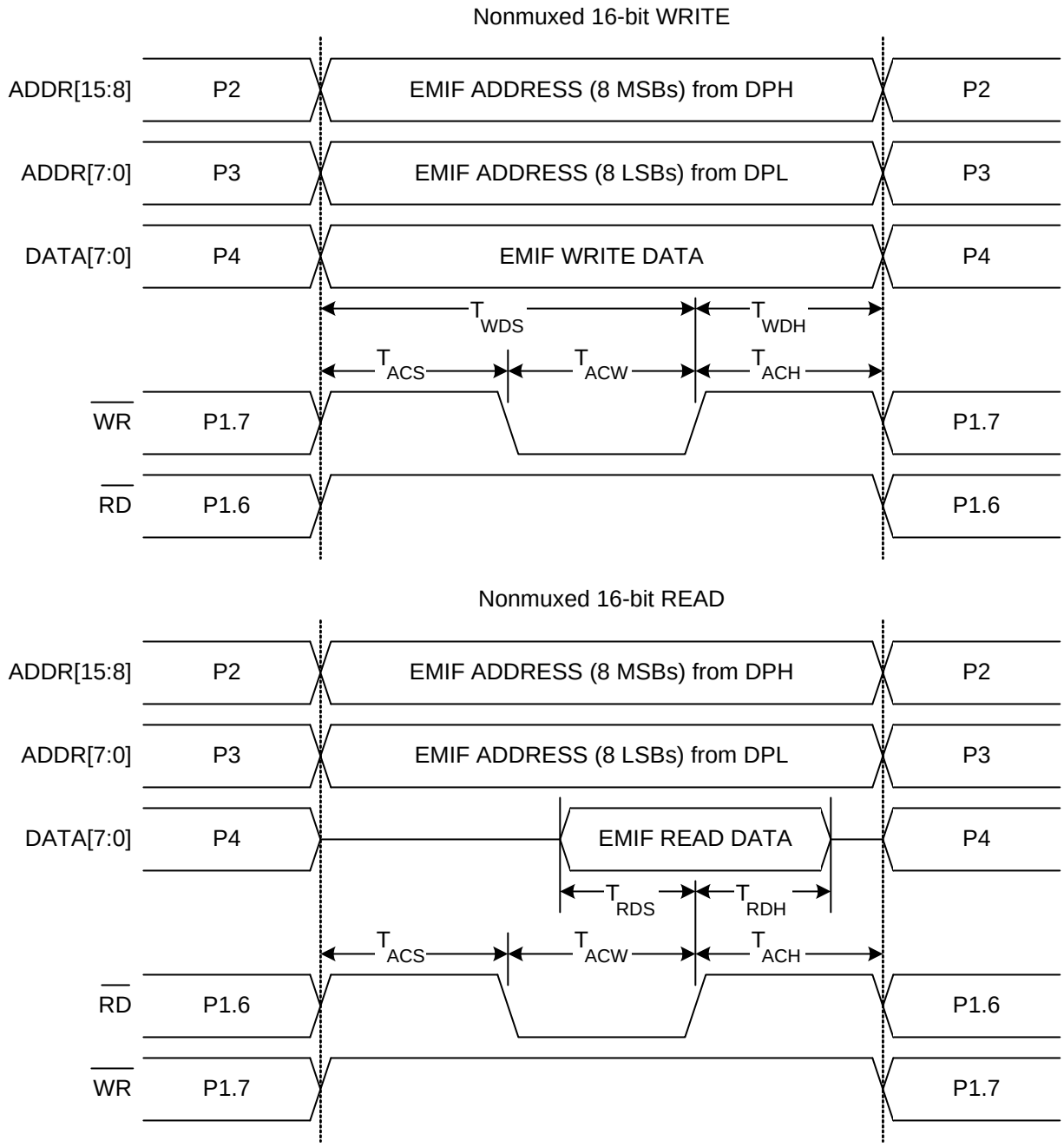

Figure 13.5. Non-multiplexed 16-bit MOVX Timing

16.4. USB Clock Configuration

USB0 is capable of communication as a Full or Low Speed USB function. Communication speed is selected via the SPEED bit in SFR USB0XCN. When operating as a Low Speed function, the USB0 clock must be 6 MHz. When operating as a Full Speed function, the USB0 clock must be 48 MHz. Clock options are described in **Section "14. Oscillators" on page 131**. The USB0 clock is selected via SFR CLKSEL (see SFR Definition 14.6).

Clock Recovery circuitry uses the incoming USB data stream to adjust the internal oscillator; this allows the internal oscillator (and 4x Clock Multiplier) to meet the requirements for USB clock tolerance. Clock Recovery should be used in the following configurations:

Communication Speed	USB Clock	4x Clock Multiplier Input
Full Speed	4x Clock Multiplier	Internal Oscillator
Low Speed	Internal Oscillator / 2	N/A

When operating USB0 as a Low Speed function with Clock Recovery, software must write '1' to the CRLOW bit to enable Low Speed Clock Recovery. Clock Recovery is typically not necessary in Low Speed mode.

Single Step Mode can be used to help the Clock Recovery circuitry to lock when high noise levels are present on the USB network. This mode is not required (or recommended) in typical USB environments.

USB Register Definition 16.5. CLKREC: Clock Recovery Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CRE	CRSSEN	CRLOW	Reserved					00001001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0F

Bit7: CRE: Clock Recovery Enable.
This bit enables/disables the USB clock recovery feature.
0: Clock recovery disabled.
1: Clock recovery enabled.

Bit6: CRSSEN: Clock Recovery Single Step.
This bit forces the oscillator calibration into 'single-step' mode during clock recovery.
0: Normal calibration mode.
1: Single step mode.

Bit5: CRLOW: Low Speed Clock Recovery Mode.
This bit must be set to '1' if clock recovery is used when operating as a Low Speed USB device.
0: Full Speed Mode.
1: Low Speed Mode.

Bits4–0: Reserved. Read = Variable. Must Write = 01001b.

Note: The USB transceiver must be enabled before enabling Clock Recovery.

Figure 17.4 shows the typical SCL generation described by Equation 17.2. Notice that T_{HIGH} is typically twice as large as T_{LOW}. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 17.1.

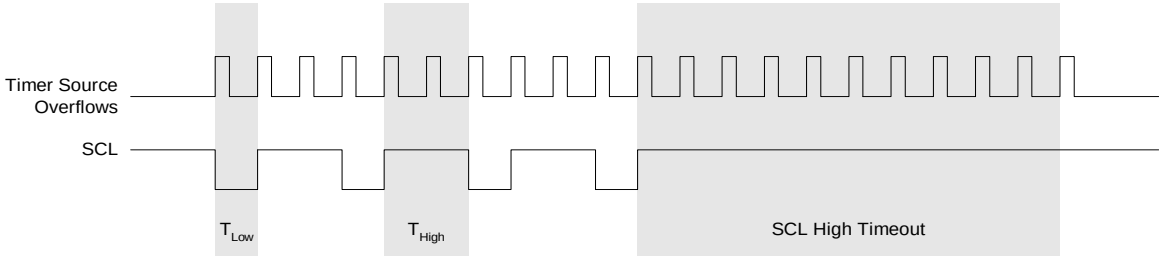

Figure 17.4. Typical SMBus SCL Generation

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 17.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 17.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	T_{low} - 4 system clocks OR 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks

***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. The s/w delay occurs between the time SMB0DAT or ACK is written and when SI 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 "17.3.3. SCL Low Timeout" on page 191**). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 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 17.4). When a Free Timeout is detected, the interface will respond as if a STOP was detected (an interrupt will be generated, and STO will be set).

17.4.3. 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 17.3. SMB0DAT: SMBus Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC2

Bits7-0: SMB0DAT: 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.

17.5. SMBus Transfer Modes

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

17.5.1. Master Transmitter Mode

Serial data is transmitted on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 17.5 shows a typical Master Transmitter sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that the ‘data byte transferred’ interrupts occur **after** the ACK cycle in this mode.

17.5.3. Slave Receiver Mode

Serial data is received on SDA and the clock is received on SCL. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. Upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. Software responds to the received slave address with an ACK, or ignores the received slave address with a NACK. If the received slave address is ignored, slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received. Software must write the ACK bit after each received byte to ACK or NACK the received byte. The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 17.7 shows a typical Slave Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

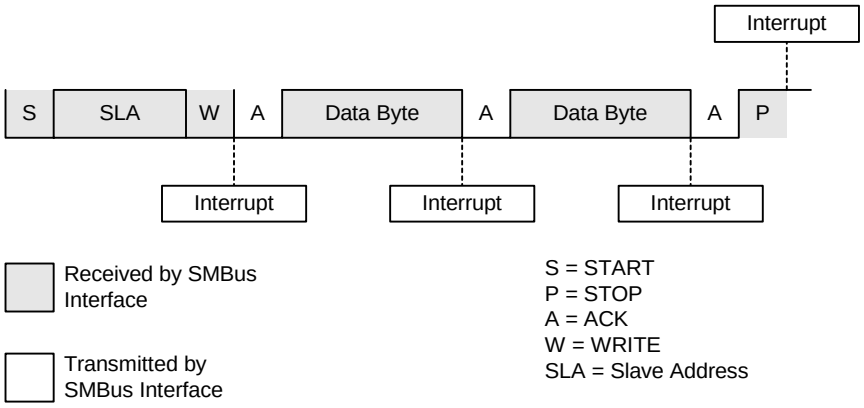

Figure 17.7. Typical Slave Receiver Sequence

19.3. Configuration and Operation

UART1 provides standard asynchronous, full duplex communication. It can operate in a point-to-point serial communications application, or as a node on a multi-processor serial interface. To operate in a point-to-point application, where there are only two devices on the serial bus, the MCE1 bit in SMOD1 should be cleared to '0'. For operation as part of a multi-processor communications bus, the MCE1 and XBE1 bits should both be set to '1'. In both types of applications, data is transmitted from the microcontroller on the TX1 pin, and received on the RX1 pin. The TX1 and RX1 pins are configured using the crossbar and the Port I/O registers, as detailed in **Section "15. Port Input/Output" on page 142**.

In typical UART communications, The transmit (TX) output of one device is connected to the receive (RX) input of the other device, either directly or through a bus transceiver, as shown in Figure 19.5.

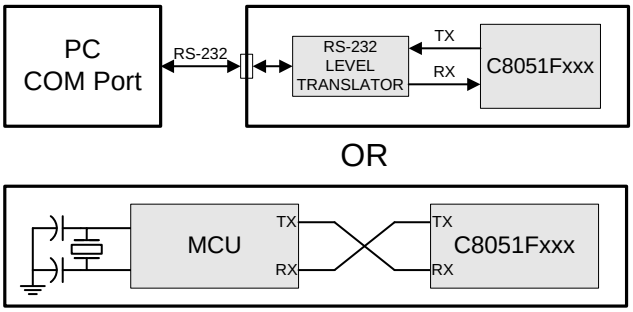

Figure 19.5. Typical UART Interconnect Diagram

19.3.1. Data Transmission

Data transmission is double-buffered, and begins when software writes a data byte to the SBUF1 register. Writing to SBUF1 places data in the Transmit Holding Register, and the Transmit Holding Register Empty flag (THRE1) will be cleared to '0'. If the UARTs shift register is empty (i.e., no transmission is in progress) the data will be placed in the shift register, and the THRE1 bit will be set to '1'. If a transmission is in progress, the data will remain in the Transmit Holding Register until the current transmission is complete. The TI1 Transmit Interrupt Flag (SCON1.1) will be set at the end of any transmission (the beginning of the stop-bit time). If enabled, an interrupt will occur when TI1 is set.

If the extra bit function is enabled (XBE1 = '1') and the parity function is disabled (PE1 = '0'), the value of the TBX1 (SCON1.3) bit will be sent in the extra bit position. When the parity function is enabled (PE1 = '1'), hardware will generate the parity bit according to the selected parity type (selected with S1PT[1:0]), and append it to the data field. Note: when parity is enabled, the extra bit function is not available.

19.3.2. Data Reception

Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to logic 1. After the stop bit is received, the data byte will be stored in the receive FIFO if the following conditions are met: the receive FIFO (3 bytes deep) must not be full, and the stop bit(s) must be logic 1. In the event that the receive FIFO is full, the incoming byte will be lost, and a Receive FIFO Overrun Error will be generated (OVR1 in register SCON1 will be set to logic 1). If the stop bit(s) were logic 0, the incoming data will not be stored in the receive FIFO. If the reception conditions are met, the data is stored in the receive FIFO, and the RI1 flag will be set. Note: when MCE1 = '1', RI1 will only be set if the extra bit was equal to '1'. Data can be read from the receive FIFO by reading the SBUF1 register. The SBUF1 register represents the oldest byte in the FIFO. After SBUF1 is read, the next byte in the FIFO is immediately loaded into SBUF1, and

Table 20.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing* (See Figure 20.8 and Figure 20.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}		ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}		ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20		ns
T_{MIH}	SCK Shift Edge to MISO Change	0		ns
Slave Mode Timing* (See Figure 20.10 and Figure 20.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}		ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}		ns
T_{SEZ}	NSS Falling to MISO Valid		4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z		4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}		ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}		ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}		ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}		ns
T_{SOH}	SCK Shift Edge to MISO Change		4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

21.2.3. Timer 2 Capture Modes: USB Start-of-Frame or LFO Falling Edge

When T2CE = ‘1’, Timer 2 will operate in one of two special capture modes. The capture event can be selected between a USB Start-of-Frame (SOF) capture, and a Low-Frequency Oscillator (LFO) Falling Edge capture, using the T2CSS bit. The USB SOF capture mode can be used to calibrate the system clock or external oscillator against the known USB host SOF clock. The LFO falling-edge capture mode can be used to calibrate the internal Low-Frequency Oscillator against the internal High-Frequency Oscillator or an external clock source. When T2SPLIT = ‘0’, Timer 2 counts up and overflows from 0xFFFF to 0x0000. Each time a capture event is received, the contents of the Timer 2 registers (TMR2H:TMR2L) are latched into the Timer 2 Reload registers (TMR2RLH:TMR2RLL). A Timer 2 interrupt is generated if enabled.

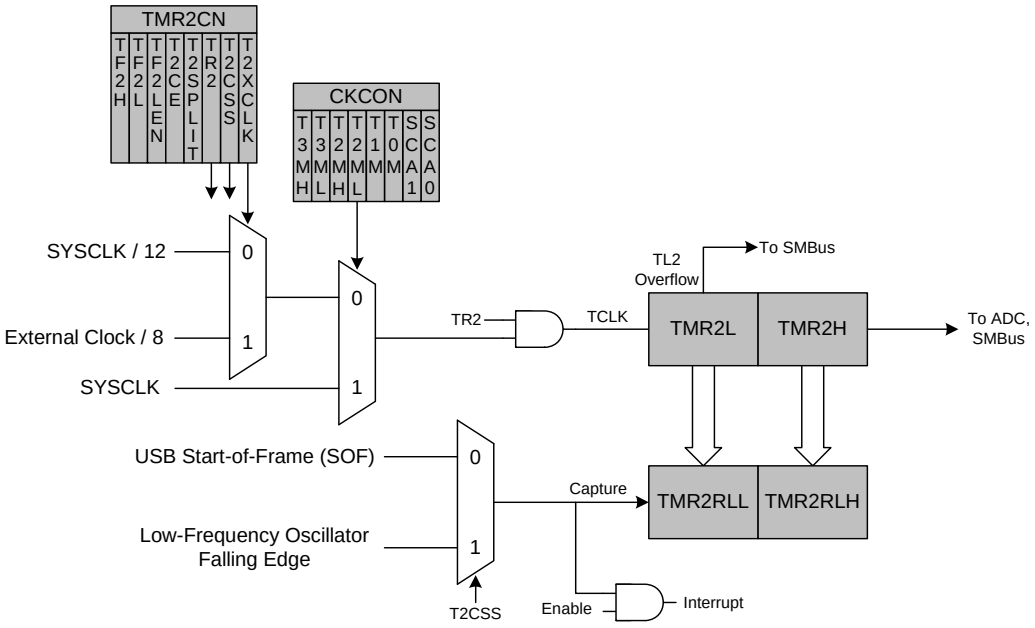

Figure 21.6. Timer 2 Capture Mode (T2SPLIT = ‘0’)

22.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. **Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter.** Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2-CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 22.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic 1). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 22.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8*

***Note:** External oscillator source divided by 8 is synchronized with the system clock.

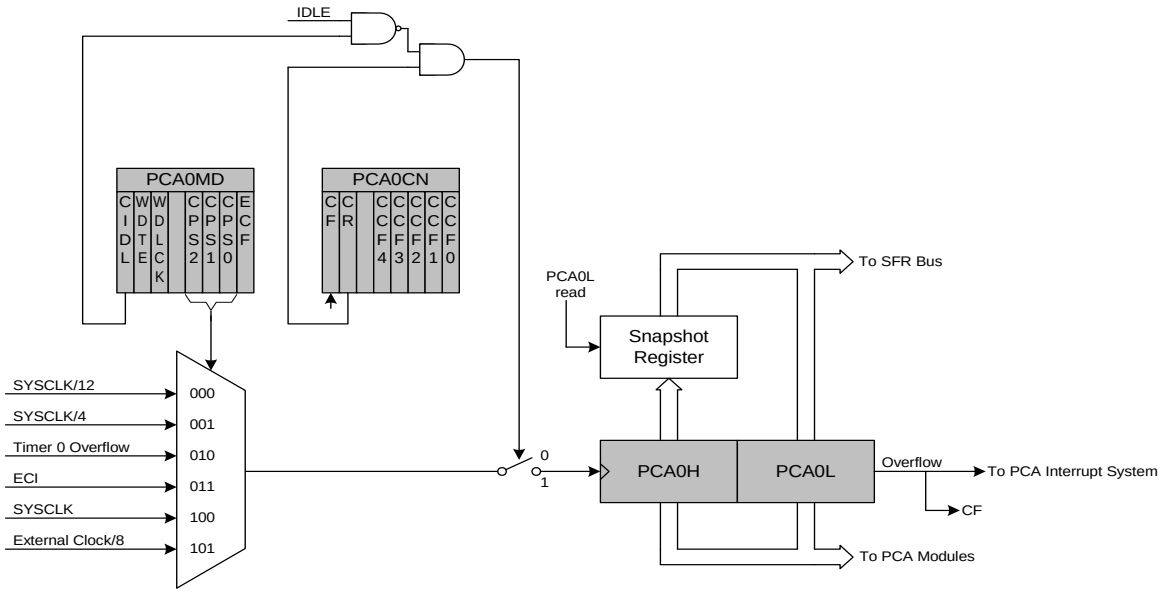

Figure 22.2. PCA Counter/Timer Block Diagram

SFR Definition 22.2. PCA0MD: PCA Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CIDL	WDTE	WDLCK	-	CPS2	CPS1	CPS0	ECF	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD9

Bit7: 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.

Bit6: 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.

Bit5: WDLCK: Watchdog Timer Lock
This bit enables and locks the Watchdog Timer. When WDLCK is set to '1', the Watchdog Timer may not be disabled until the next system reset.
0: Watchdog Timer unlocked.
1: Watchdog Timer enabled and locked.

Bit4: UNUSED. Read = 0b, Write = don't care.

Bits3–1: CPS2–CPS0: PCA Counter/Timer Pulse Select.
These bits select the timebase source for the PCA counter.

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External clock divided by 8*
1	1	0	Reserved
1	1	1	Reserved

***Note:** External oscillator source divided by 8 is synchronized with the system clock.

Bit0: 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 PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.

SFR Definition 22.4. PCA0L: PCA Counter/Timer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xF9

Bits 7–0: PCA0L: PCA Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

SFR Definition 22.5. PCA0H: PCA Counter/Timer High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xFA

Bits 7–0: PCA0H: PCA Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer.

SFR Definition 22.6. PCA0CPLn: PCA Capture Module Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xFB, 0xE9, 0xEB, 0xED, 0xFD

PCA0CPLn Address: PCA0CPL0 = 0xFB (n = 0), PCA0CPL1 = 0xE9 (n = 1),
PCA0CPL2 = 0xEB (n = 2), PCA0CPL3 = 0xED (n = 3),
PCA0CPL4 = 0xFD (n = 4)

Bits7–0: PCA0CPLn: PCA Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.