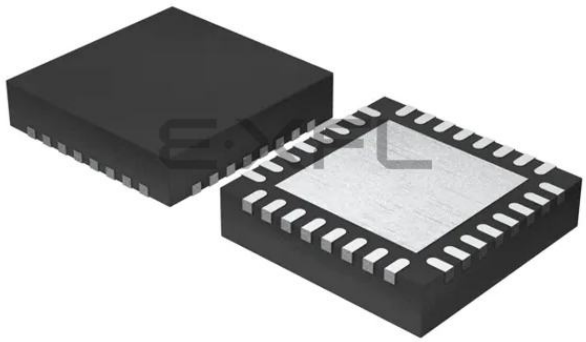

Welcome to E-XFL.COM

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	SMBus (2-Wire/I²C), SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	25
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	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f347-gmr

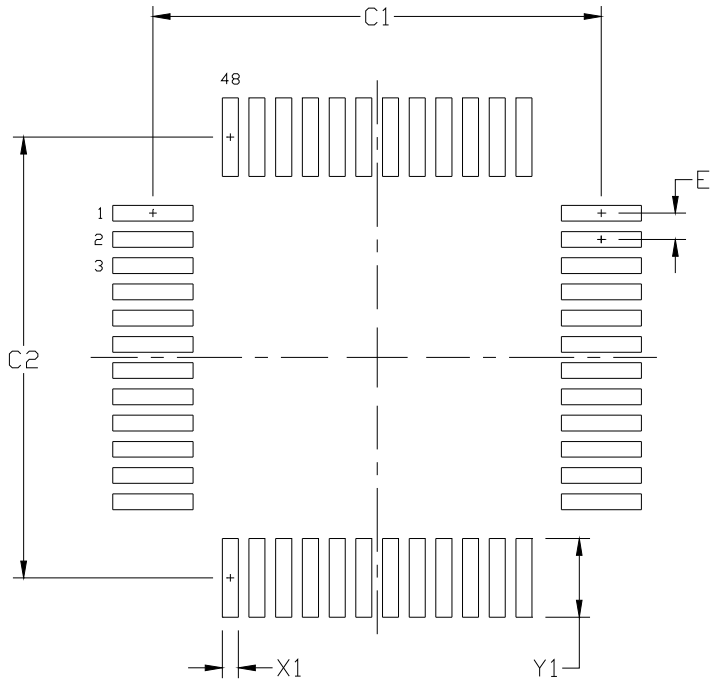

Figure 4.3. TQFP-48 Recommended PCB Land Pattern

Table 4.3. TQFP-48 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	8.30	8.40
C2	8.30	8.40
E	0.50 BSC	
X1	0.20	0.30
Y1	1.40	1.50

Notes:

General:

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

Solder Mask Design:

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

Stencil Design:

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

Card Assembly:

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

Table 7.1. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +85 °C unless otherwise noted.
All specifications apply to both Comparator0 and Comparator1 unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		100		ns
	CP0+ – CP0– = –100 mV		250		ns
Response Time: Mode 1, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		175		ns
	CP0+ – CP0– = –100 mV		500		ns
Response Time: Mode 2, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		320		ns
	CP0+ – CP0– = –100 mV		1100		ns
Response Time: Mode 3, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		1050		ns
	CP0+ – CP0– = –100 mV		5200		ns
Common-Mode Rejection Ratio			1.5	4	mV/V
Positive Hysteresis 1	CP0HYP1–0 = 00		0	1	mV
Positive Hysteresis 2	CP0HYP1–0 = 01	2	5	10	mV
Positive Hysteresis 3	CP0HYP1–0 = 10	7	10	20	mV
Positive Hysteresis 4	CP0HYP1–0 = 11	15	20	30	mV
Negative Hysteresis 1	CP0HYN1–0 = 00		0	1	mV
Negative Hysteresis 2	CP0HYN1–0 = 01	2	5	10	mV
Negative Hysteresis 3	CP0HYN1–0 = 10	7	10	20	mV
Negative Hysteresis 4	CP0HYN1–0 = 11	15	20	30	mV
Inverting or Non-Inverting Input Voltage Range		–0.25		V_{DD} + 0.25	V
Input Capacitance			3		pF
Input Bias Current			0.001		nA
Input Offset Voltage		–5		+5	mV
Power Supply					
Power Supply Rejection			0.1		mV/V
Power-up Time			10		µs
Supply Current at DC	Mode 0		7.6		µA
	Mode 1		3.2		µA
	Mode 2		1.3		µA
	Mode 3		0.4		µA

***Note:** Vcm is the common-mode voltage on CP0+ and CP0–.

13. External Data Memory Interface and On-Chip XRAM

4k Bytes (C8051F340/2/4/6/A/C/D) or 2k Bytes (C8051F341/3/5/7/8/9/B) of RAM are included on-chip, and mapped into the external data memory space (XRAM). The 1k Bytes of USB FIFO space can also be mapped into XRAM address space for additional general-purpose data storage. Additionally, an External Memory Interface (EMIF) is available on the C8051F340/1/4/5/8/C devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 13.1). Note: the MOVX instruction can also be used for writing to the FLASH memory. See **Section “12. Flash Memory” on page 107** for details. The MOVX instruction accesses XRAM by default.

13.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

13.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

13.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

13.5. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMI0CF.4) bit.

13.5.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 13.2.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the 'Q' outputs reflect the states of the 'D' inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time $\overline{RD}$ or $\overline{WR}$ is asserted.

See **Section "13.7.2. Multiplexed Mode" on page 127** for more information.

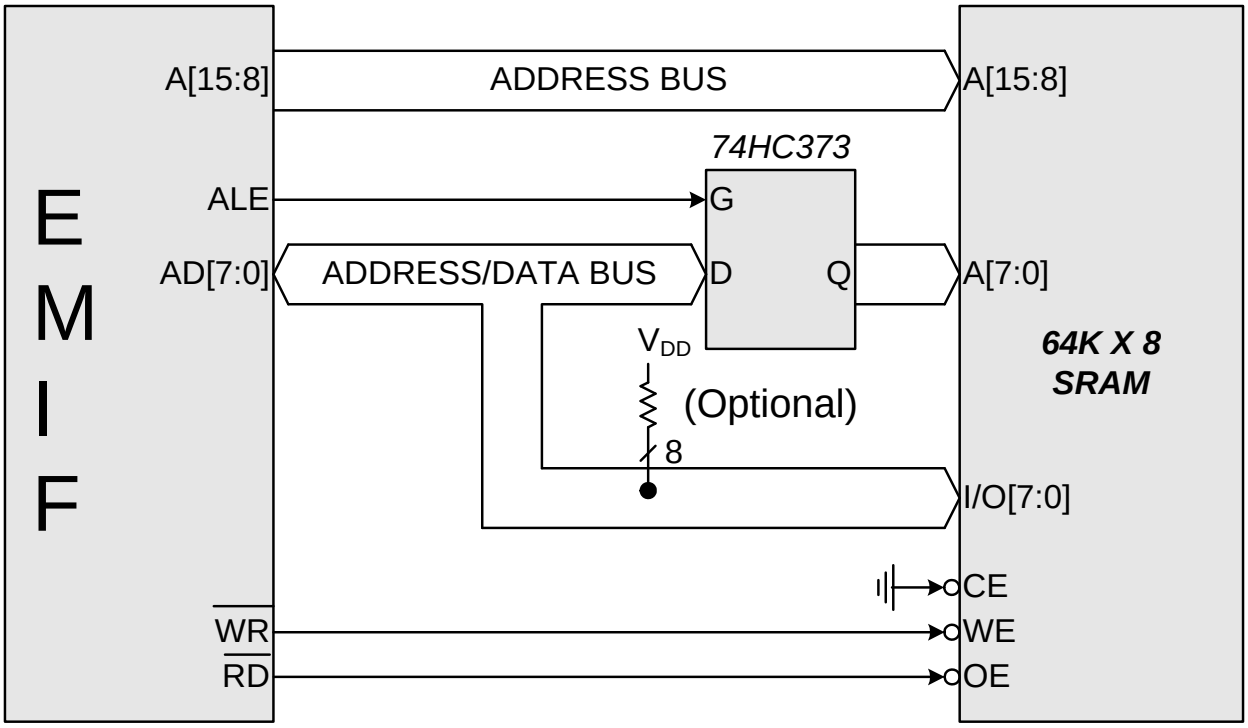

Figure 13.2. Multiplexed Configuration Example

13.7.1. Non-multiplexed Mode

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

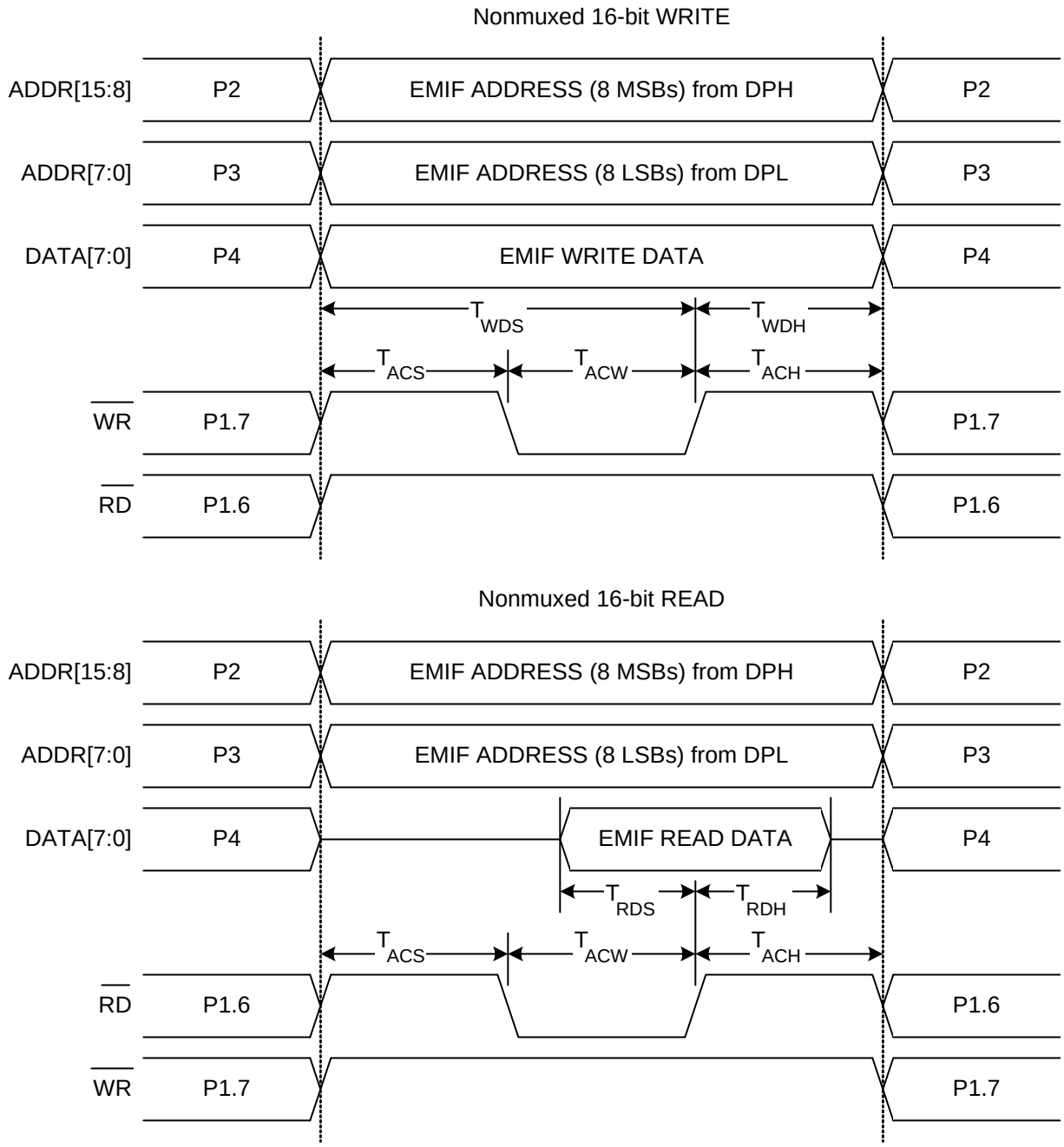

Figure 13.5. Non-multiplexed 16-bit MOVX Timing

15.2. Port I/O Initialization

Port I/O initialization consists of the following steps:

Step 1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
Step 2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
Step 3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
Step 4. Assign Port pins to desired peripherals (XBR0, XBR1).
Step 5. Enable the Crossbar (XBARE = ‘1’).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pull-up, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however this practice is not recommended. To configure a Port pin for digital input, write ‘0’ to the corresponding bit in register PnMDOUT, and write ‘1’ to the corresponding Port latch (register Pn).

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a ‘1’ indicates a digital input, and a ‘0’ indicates an analog input. All pins default to digital inputs on reset.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR1 is ‘0’, a weak pull-up is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pull-up is turned off on an output that is driving a ‘0’ to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to ‘1’ enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility of the Silicon Labs IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

Important Note: The Crossbar must be enabled to use Ports P0, P1, P2, and P3 as standard Port I/O in output mode. These Port output drivers are disabled while the Crossbar is disabled. Port 4 always functions as standard GPIO.

SFR Definition 15.16. P3: Port3 Latch

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB0

Bits7–0: P3.[7:0]
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P3MDOUT.n bit = 0).
Read - Always reads '0' if selected as analog input in register P3MDIN. Directly reads Port pin when configured as digital input.
0: P3.n pin is logic low.
1: P3.n pin is logic high.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.17. P3MDIN: Port3 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xF4

Bits7–0: Analog Input Configuration Bits for P3.7–P3.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, digital driver, and digital receiver disabled.
0: Corresponding P3.n pin is configured as an analog input.
1: Corresponding P3.n pin is not configured as an analog input.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.18. P3MDOUT: Port3 Output Mode

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:
								0xA7

Bits7–0: Output Configuration Bits for P3.7–P3.0 (respectively); ignored if corresponding bit in register P3MDIN is logic 0.
0: Corresponding P3.n Output is open-drain.
1: Corresponding P3.n Output is push-pull.

Note: P3.1–3.7 are only available on 48-pin devices.

USB Register Definition 16.16. CMIE: USB0 Common Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	SOFE	RSTINTE	RSUINTE	SUSINTE	00000110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0B

Bits7–4: Unused. Read = 0000b; Write = don't care.
Bit3: SOFE: Start of Frame Interrupt Enable
0: SOF interrupt disabled.
1: SOF interrupt enabled.
Bit2: RSTINTE: Reset Interrupt Enable
0: Reset interrupt disabled.
1: Reset interrupt enabled.
Bit1: RSUINTE: Resume Interrupt Enable
0: Resume interrupt disabled.
1: Resume interrupt enabled.
Bit0: SUSINTE: Suspend Interrupt Enable
0: Suspend interrupt disabled.
1: Suspend interrupt enabled.

16.9. The Serial Interface Engine

The Serial Interface Engine (SIE) performs all low level USB protocol tasks, interrupting the processor when data has successfully been transmitted or received. When receiving data, the SIE will interrupt the processor when a complete data packet has been received; appropriate handshaking signals are automatically generated by the SIE. When transmitting data, the SIE will interrupt the processor when a complete data packet has been transmitted and the appropriate handshake signal has been received.

The SIE will not interrupt the processor when corrupted/erroneous packets are received.

16.10. Endpoint0

Endpoint0 is managed through the USB register E0CSR (USB Register Definition 16.17). The INDEX register must be loaded with 0x00 to access the E0CSR register.

An Endpoint0 interrupt is generated when:

1. A data packet (OUT or SETUP) has been received and loaded into the Endpoint0 FIFO. The OPRDY bit (E0CSR.0) is set to '1' by hardware.
2. An IN data packet has successfully been unloaded from the Endpoint0 FIFO and transmitted to the host; INPRDY is reset to '0' by hardware.
3. An IN transaction is completed (this interrupt generated during the status stage of the transaction).
4. Hardware sets the STSTL bit (E0CSR.2) after a control transaction ended due to a protocol violation.
5. Hardware sets the SUEND bit (E0CSR.4) because a control transfer ended before firmware sets the DATAEND bit (E0CSR.3).

USB Register Definition 16.19. EINCSRL: USB0 IN Endpoint Control Low Byte

R	W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	CLRDT	STSTL	SDSTL	FLUSH	UNDRUN	FIFONE	INPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x11

Bit7: Unused. Read = 0; Write = don't care.

Bit6: CLRDT: Clear Data Toggle.
Write: Software should write '1' to this bit to reset the IN Endpoint data toggle to '0'.
Read: This bit always reads '0'.

Bit5: STSTL: Sent Stall
Hardware sets this bit to '1' when a STALL handshake signal is transmitted. The FIFO is flushed, and the INPRDY bit cleared. This flag must be cleared by software.

Bit4: SDSTL: Send Stall.
Software should write '1' to this bit to generate a STALL handshake in response to an IN token. Software should write '0' to this bit to terminate the STALL signal. This bit has no effect in ISO mode.

Bit3: FLUSH: FIFO Flush.
Writing a '1' to this bit flushes the next packet to be transmitted from the IN Endpoint FIFO. The FIFO pointer is reset and the INPRDY bit is cleared. If the FIFO contains multiple packets, software must write '1' to FLUSH for each packet. Hardware resets the FLUSH bit to '0' when the FIFO flush is complete.

Bit2: UNDRUN: Data Underrun.
The function of this bit depends on the IN Endpoint mode:
Isochronous: Set when a zero-length packet is sent after an IN token is received while bit INPRDY = '0'.
Interrupt/Bulk: This bit is not used in these modes and will always read a '0'.
This bit must be cleared by software.

Bit1: FIFONE: FIFO Not Empty.
0: The IN Endpoint FIFO is empty.
1. The IN Endpoint FIFO contains one or more packets.

Bit0: INPRDY: In Packet Ready.
Software should write '1' to this bit after loading a data packet into the IN Endpoint FIFO. Hardware clears INPRDY due to any of the following:
1. A data packet is transmitted.
2. Double buffering is enabled (DBIEN = '1') and there is an open FIFO packet slot.
3. If the endpoint is in Isochronous Mode (ISO = '1') and ISOUD = '1', INPRDY will read '0' until the next SOF is received.
An interrupt (if enabled) will be generated when hardware clears INPRDY as a result of a packet being transmitted.

USB Register Definition 16.21. EOUTCSRL: USB0 OUT Endpoint Control Low Byte

W	R/W	R/W	R/W	R	R/W	R	R/W	Reset Value
CLRDT	STSTL	SDSTL	FLUSH	DATERR	OVRUN	FIFOFUL	OPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x14

Bit7: CLRDT: Clear Data Toggle
Write: Software should write ‘1’ to this bit to reset the OUT endpoint data toggle to ‘0’.
Read: This bit always reads ‘0’.

Bit6: STSTL: Sent Stall
Hardware sets this bit to ‘1’ when a STALL handshake signal is transmitted. This flag must be cleared by software.

Bit5: SDSTL: Send Stall
Software should write ‘1’ to this bit to generate a STALL handshake. Software should write ‘0’ to this bit to terminate the STALL signal. This bit has no effect in ISO mode.

Bit4: FLUSH: FIFO Flush
Writing a ‘1’ to this bit flushes the next packet to be read from the OUT endpoint FIFO. The FIFO pointer is reset and the OPRDY bit is cleared. If the FIFO contains multiple packets, software must write ‘1’ to FLUSH for each packet. Hardware resets the FLUSH bit to ‘0’ when the FIFO flush is complete.

Note: If data for the current packet has already been read from the FIFO, the FLUSH bit should not be used to flush the packet. Instead, the entire data packet should be read from the FIFO manually.

Bit3: DATERR: Data Error
In ISO mode, this bit is set by hardware if a received packet has a CRC or bit-stuffing error. It is cleared when software clears OPRDY. This bit is only valid in ISO mode.

Bit2: OVRUN: Data Overrun
This bit is set by hardware when an incoming data packet cannot be loaded into the OUT endpoint FIFO. This bit is only valid in ISO mode, and must be cleared by software.
0: No data overrun.
1: A data packet was lost because of a full FIFO since this flag was last cleared.

Bit1: FIFOFUL: OUT FIFO Full
This bit indicates the contents of the OUT FIFO. If double buffering is enabled for the endpoint (DBIEN = ‘1’), the FIFO is full when the FIFO contains two packets. If DBIEN = ‘0’, the FIFO is full when the FIFO contains one packet.
0: OUT endpoint FIFO is not full.
1: OUT endpoint FIFO is full.

Bit0: OPRDY: OUT Packet Ready
Hardware sets this bit to ‘1’ and generates an interrupt when a data packet is available. Software should clear this bit after each data packet is unloaded from the OUT endpoint FIFO.

Table 16.4. USB Transceiver Electrical Characteristics

V_{DD} = 3.0 to 3.6 V, –40 to +85 °C unless otherwise specified

Parameters	Symbol	Conditions	Min	Typ	Max	Units
Transmitter						
Output High Voltage	V_{OH}		2.8			V
Output Low Voltage	V_{OL}				0.8	V
Output Crossover Point	V_{CRS}		1.3		2.0	V
Output Impedance	Z_{DRV}	Driving High Driving Low		38 38		Ω
Pull-up Resistance	R_{PU}	Full Speed (D+ Pull-up) Low Speed (D– Pull-up)	1.425	1.5	1.575	kΩ
Output Rise Time	T_R	Low Speed Full Speed	75 4		300 20	ns
Output Fall Time	T_F	Low Speed Full Speed	75 4		300 20	ns
Receiver						
Differential Input Sensitivity	V_{DI}	| (D+) – (D–) |	0.2			V
Differential Input Common Mode Range	V_{CM}		0.8		2.5	V
Input Leakage Current	I_L	Pullups Disabled		<1.0		µA

Note: Refer to the USB Specification for timing diagrams and symbol definitions.

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.

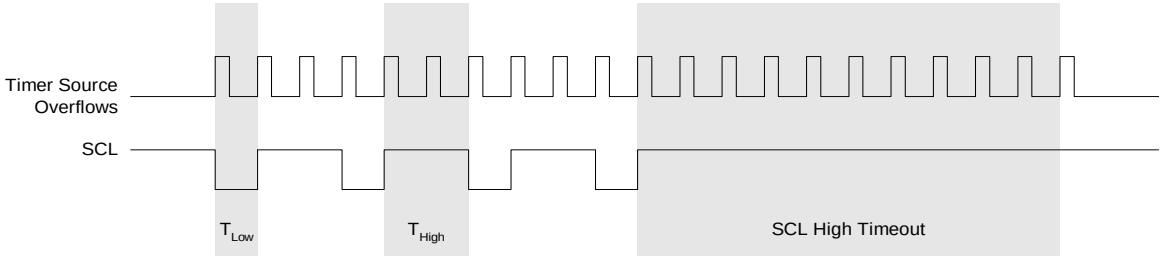

Figure 17.4. Typical SMBus SCL Generation

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).

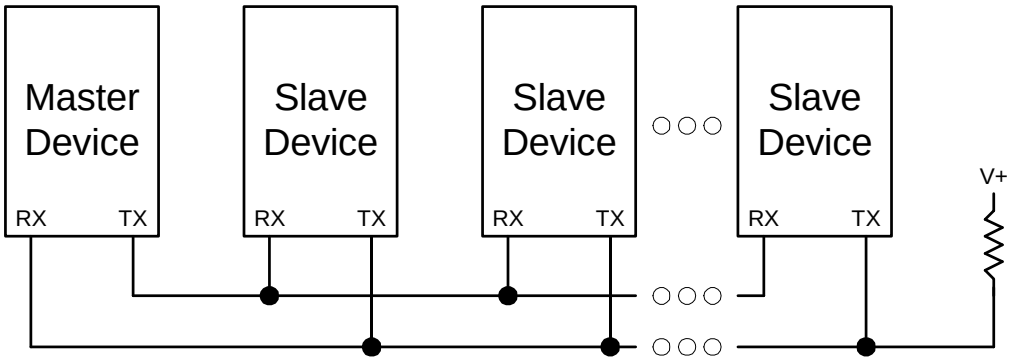

Figure 18.6. UART Multi-Processor Mode Interconnect Diagram

20.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 20.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 20.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

20.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

Note that all of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

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

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

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

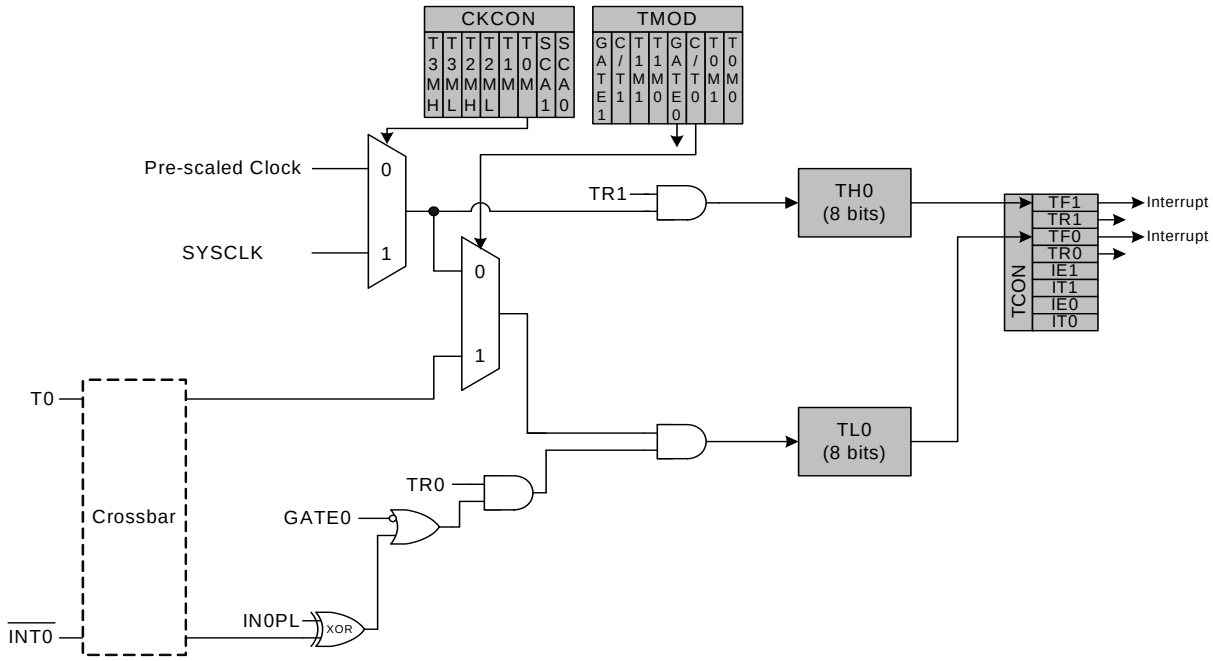

Figure 21.3. T0 Mode 3 Block Diagram

SFR Definition 21.2. TMOD: Timer Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x89

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{INT1}$ logic level.
1: Timer 1 enabled only when TR1 = 1 AND $\overline{INT1}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.3).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{INT0}$ logic level.
1: Timer 0 enabled only when TR0 = 1 AND $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.2).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

21.3. 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, (split) 8-bit auto-reload mode, USB Start-of-Frame (SOF) capture mode, or Low-Frequency Oscillator (LFO) Rising Edge capture mode. The Timer 3 operation mode is defined by the T3SPLIT (TMR3CN.3), T3CE (TMR3CN.4) bits, and T3CSS (TMR3CN.1) bits.

Timer 3 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 3 (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.

21.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is ‘0’ and T3CE = ‘0’, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 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 3 reload registers (TMR3RLH and TM3RLL) is loaded into the Timer 3 register as shown in Figure 21.4, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled, an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

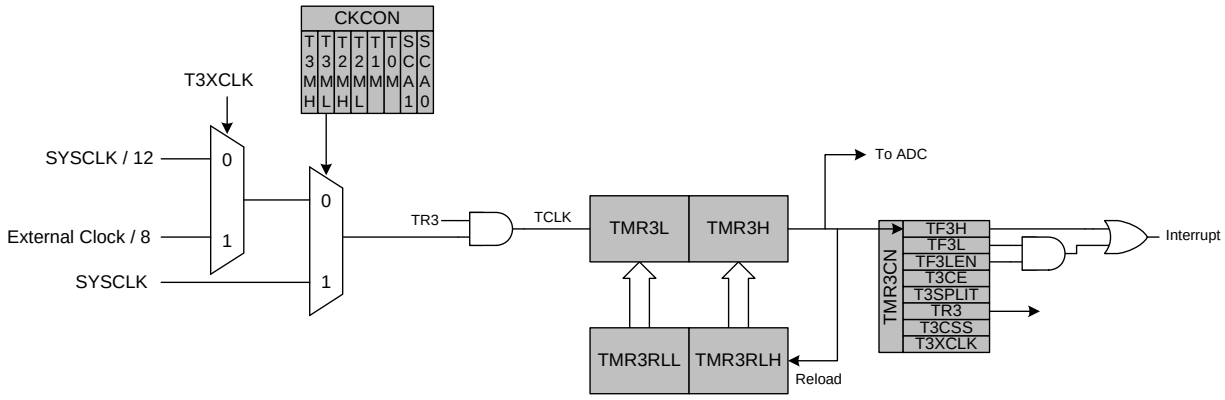

Figure 21.8. Timer 3 16-Bit Mode Block Diagram

21.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is ‘1’ and T3CE = ‘0’, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 21.5. 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, or the external oscillator clock source divided by 8. 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 bit (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK / 12
0	1	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.

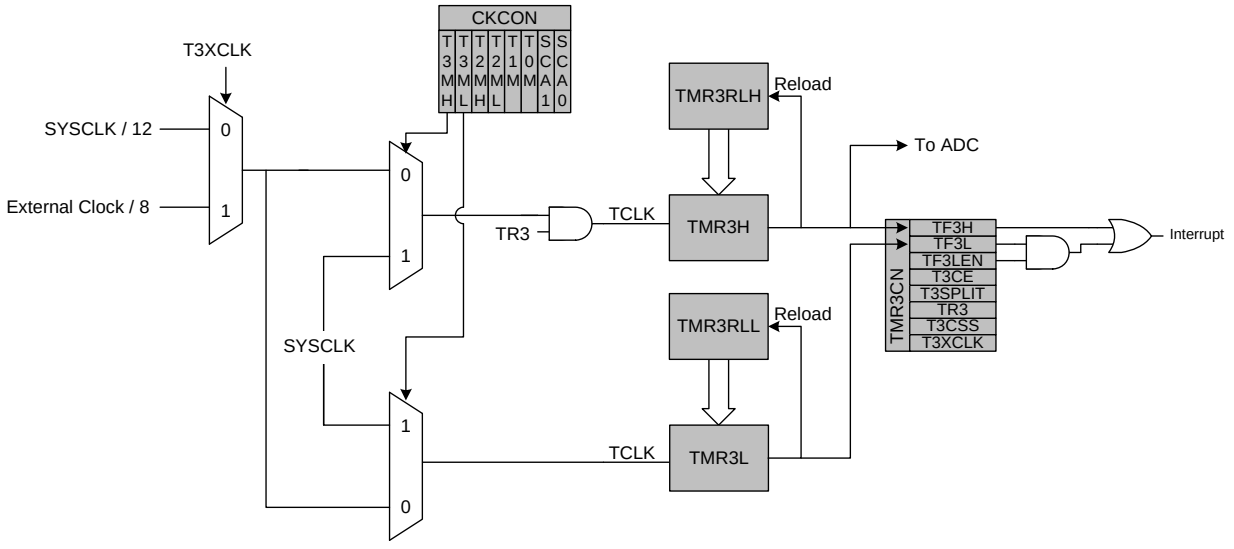

Figure 21.9. Timer 3 8-Bit Mode Block Diagram

$$Offset = (256 \times PCA0CPL4) + (256 - PCA0L)$$

Equation 22.4. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH4 and PCA0H. Software may force a WDT reset by writing a '1' to the CCF4 flag (PCA0CN.4) while the WDT is enabled.

22.3.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

1. Disable the WDT by writing a '0' to the WDTE bit.
2. Select the desired PCA clock source (with the CPS2-CPS0 bits).
3. Load PCA0CPL4 with the desired WDT update offset value.
4. Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
5. Enable the WDT by setting the WDTE bit to '1'.
6. (optional) Lock the WDT (prevent WDT disable until the next system reset) by setting the WDLCK bit to '1'.
7. Write a value to PCA0CPH4 to reload the WDT.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL4 defaults to 0x00. Using Equation 22.4, this results in a WDT timeout interval of 256 PCA clocks. Table 22.3 lists some example timeout intervals for typical system clocks.

Table 22.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL4	Timeout Interval (ms)
12,000,000	255	65.5
12,000,000	128	33.0
12,000,000	32	8.4
24,000,000	255	32.8
24,000,000	128	16.5
24,000,000	32	4.2
1,500,000[2]	255	524.3
1,500,000[2]	128	264.2
1,500,000[2]	32	67.6
32,768	255	24,000
32,768	128	12,093.75
32,768	32	3,093.75

Notes:

1. Assumes SYSCLK / 12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. System Clock reset frequency.

23. C2 Interface

C8051F34x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

23.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 23.1. C2ADD: C2 Address

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

Address	Description
0x00	Selects the Device ID register for Data Read instructions
0x01	Selects the Revision ID register for Data Read instructions
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
0xAD	Selects the C2 Flash Programming Data register for Data Read/Write instructions

C2 Register Definition 23.2. DEVICEID: C2 Device ID

Reset Value: 00001111

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit device ID: 0x0F (C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D).