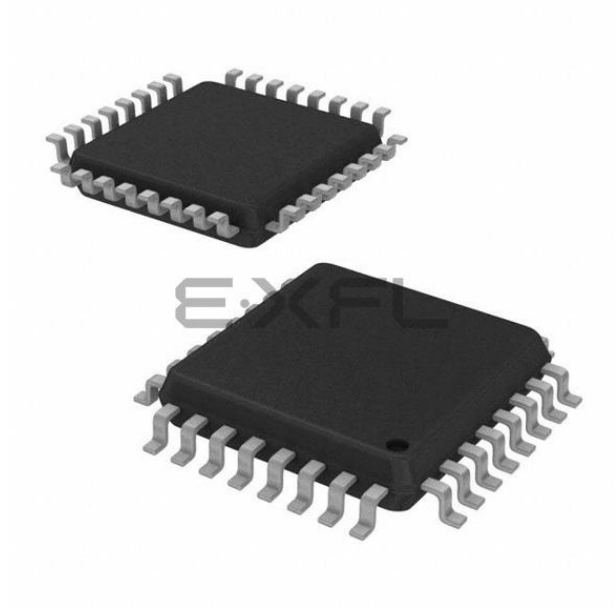

Welcome to E-XFL.COM

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	48MHz
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	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 17x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f34a-gqr

3. Global DC Electrical Characteristics

Table 3.1. Global DC Electrical Characteristics

–40 to +85 °C, 25 MHz System Clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Voltage[1]		VRST	3.3	3.6	V
Digital Supply RAM Data Retention Voltage			1.5		V
SYSCLK (System Clock)[2]	C8051F340/1/2/3/A/B/C/D C8051F344/5/6/7/8/9	0 0		48 25	MHz
Specified Operating Temperature Range		–40		+85	°C
Digital Supply Current - CPU Active (Normal Mode, accessing Flash)					
I_{DD}[3]	V_{DD} = 3.3 V, SYSCLK = 48 MHz V_{DD} = 3.3 V, SYSCLK = 24 MHz V_{DD} = 3.3 V, SYSCLK = 1 MHz V_{DD} = 3.3 V, SYSCLK = 80 kHz V_{DD} = 3.6 V, SYSCLK = 48 MHz V_{DD} = 3.6 V, SYSCLK = 24 MHz		25.9 13.9 0.69 55 29.7 15.9	28.5 15.7 32.3 18	mA mA mA µA mA mA
I_{DD} Supply Sensitivity[3,4]	SYSCLK = 1 MHz, relative to V_{DD} = 3.3 V SYSCLK = 24 MHz, relative to V_{DD} = 3.3 V		47 46		%/V %/V
I_{DD} Frequency Sensitivity[3,5]	V_{DD} = 3.3 V, SYSCLK ≤ 30 MHz, T = 25 ºC V_{DD} = 3.3 V, SYSCLK > 30 MHz, T = 25 ºC V_{DD} = 3.6 V, SYSCLK ≤ 30 MHz, T = 25 ºC V_{DD} = 3.6 V, SYSCLK > 30 MHz, T = 25 ºC		0.69 0.44 0.80 0.50		mA/MHz mA/MHz mA/MHz mA/MHz
Digital Supply Current - CPU Inactive (Idle Mode, not accessing Flash)					
I_{DD}[3]	V_{DD} = 3.3 V, SYSCLK = 48 MHz V_{DD} = 3.3 V, SYSCLK = 24 MHz V_{DD} = 3.3 V, SYSCLK = 1 MHz V_{DD} = 3.3 V, SYSCLK = 80 kHz V_{DD} = 3.6 V, SYSCLK = 48 MHz V_{DD} = 3.6 V, SYSCLK = 24 MHz		16.6 8.25 0.44 35 18.6 9.26	18.75 9.34 20.9 10.5	mA mA mA µA mA mA
I_{DD} Supply Sensitivity[3,4]	SYSCLK = 1 MHz, relative to V_{DD} = 3.3 V SYSCLK = 24 MHz, relative to V_{DD} = 3.3 V		41 39		%/V %/V

Table 3.2. Index to Electrical Characteristics Tables

Table Title	Page No.
ADC0 Electrical Characteristics	56
Voltage Reference Electrical Characteristics	58
Comparator Electrical Characteristics	68
Voltage Regulator Electrical Specifications	69
Reset Electrical Characteristics	106
Flash Electrical Characteristics	109
AC Parameters for External Memory Interface	130
Oscillator Electrical Characteristics	141
Port I/O DC Electrical Characteristics	158
USB Transceiver Electrical Characteristics	187

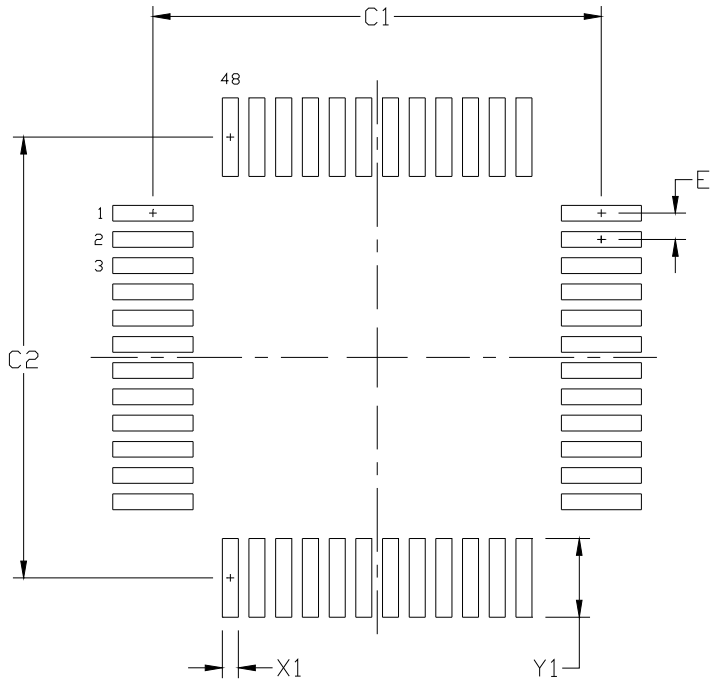

Figure 4.3. TQFP-48 Recommended PCB Land Pattern

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.

5.3.3. Settling Time Requirements

When the ADC0 input configuration is changed (i.e., a different AMUX0 selection is made), a minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the minimum tracking time requirements.

Figure 5.5 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 5.1. When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 5.1 for ADC0 minimum settling time requirements.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 5.1. ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (10).

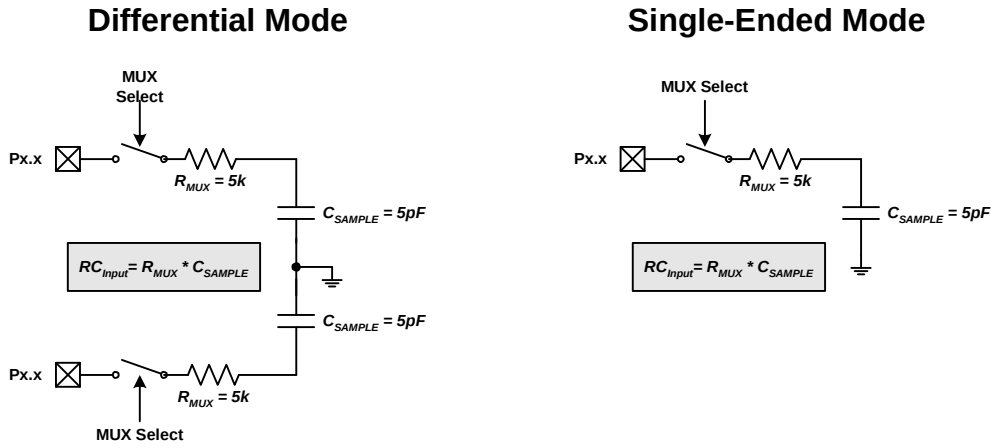

Figure 5.5. ADC0 Equivalent Input Circuits

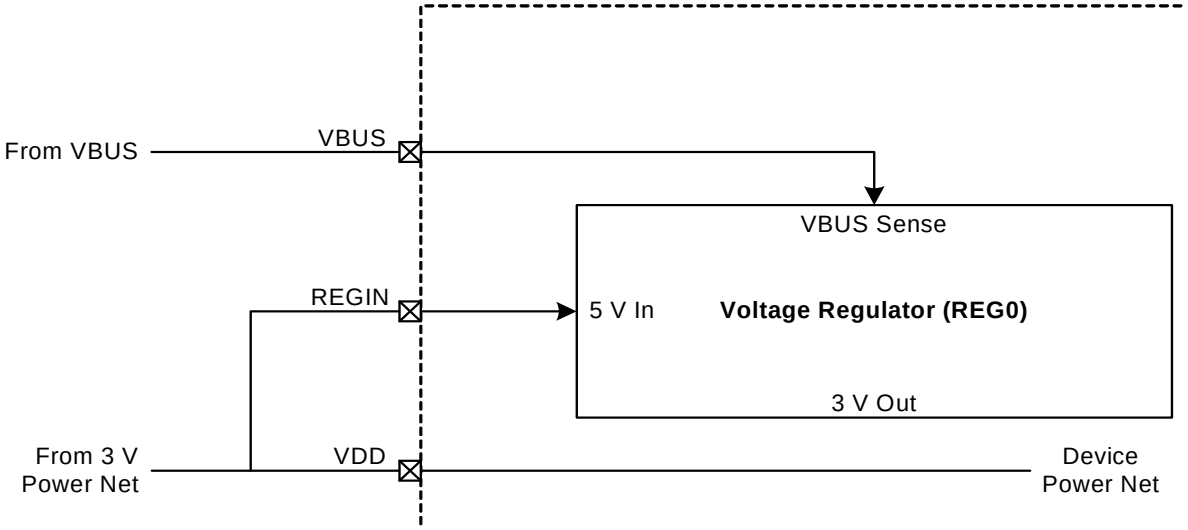

Figure 8.3. REG0 Configuration: USB Self-Powered, Regulator Disabled

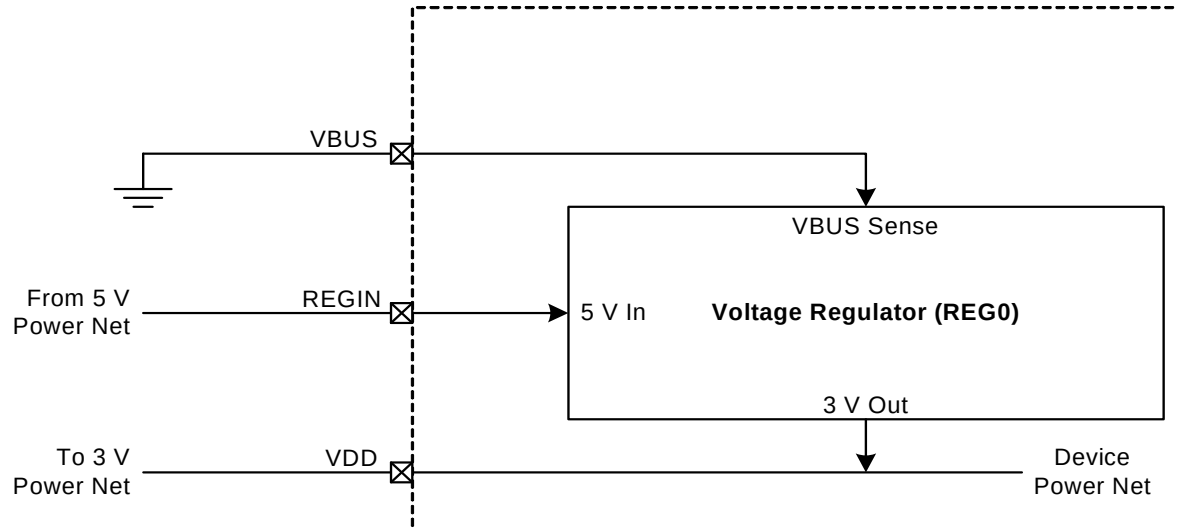

Figure 8.4. REG0 Configuration: No USB Connection

SFR Definition 9.4. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xD0

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: RS1–RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00 - 0x07
0	1	1	0x08 - 0x0F
1	0	2	0x10 - 0x17
1	1	3	0x18 - 0x1F

Bit2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
- An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
- A MUL instruction results in an overflow (result is greater than 255).
- A DIV instruction causes a divide-by-zero condition.

The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

SFR Definition 9.5. ACC: Accumulator

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE0

Bits7–0: ACC: Accumulator.
This register is the accumulator for arithmetic operations.

Table 11.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{\text{RST}}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.7 to 3.6 V			0.6	V
$\overline{\text{RST}}$ Input High Voltage		0.7 x V_{DD}			V
$\overline{\text{RST}}$ Input Low Voltage				0.3 x V_{DD}	
$\overline{\text{RST}}$ Input Pull-Up Current	$\overline{\text{RST}}$ = 0.0 V		25	40	µA
V_{DD} POR Threshold (V_{RST})		2.40	2.55	2.70	V
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	220	500	µs
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	5.0			µs
Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset		15			µs
V_{DD} Monitor Turn-on Time		100			µs
V_{DD} Monitor Supply Current			20	50	µA

12. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation. Refer to Table 12.1 for complete Flash memory electrical characteristics.

12.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see **Section “23. C2 Interface” on page 271**.

To ensure the integrity of Flash contents, it is strongly recommended that the V_{DD} monitor be left enabled in any system which writes or erases Flash memory from code. It is also crucial to ensure that the FLRT bit in register FLSCL be set to '1' if a clock speed higher than 25 MHz is being used for the device.

12.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 12.2.

12.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY); and (2) Setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed must be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE bit (register PSCTL).
Step 8. Clear the PSEE bit (register PSCTI).

12.1.3. Flash Write Procedure

Bytes in Flash memory can be written one byte at a time, or in groups of two. The FLBWE bit in register PFE0CN (SFR Definition 10.1) controls whether a single byte or a block of two bytes is written to Flash during a write operation. When FLBWE is cleared to ‘0’, the Flash will be written one byte at a time. When FLBWE is set to ‘1’, the Flash will be written in two-byte blocks. Block writes are performed in the same amount of time as single-byte writes, which can save time when storing large amounts of data to Flash memory.During a single-byte write to Flash, bytes are written individually, and a Flash write will be performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is:

Step 1. Disable interrupts.
Step 2. Clear the FLBWE bit (register PFE0CN) to select single-byte write mode.
Step 3. Set the PSWE bit (register PSCTL).
Step 4. Clear the PSEE bit (register PSCTL).
Step 5. Write the first key code to FLKEY: 0xA5.
Step 6. Write the second key code to FLKEY: 0xF1.
Step 7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
Step 8. Clear the PSWE bit.
Step 9. Re-enable interrupts.

Steps 5-7 must be repeated for each byte to be written.

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. A Flash write block is two bytes long, from even addresses to odd addresses. Writes must be performed sequentially (i.e. addresses ending in 0b and 1b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 1b. If a byte in the block does not need to be updated in Flash, it should be written to 0xFF. The recommended procedure for writing Flash in blocks is:

Step 1. Disable interrupts.
Step 2. Set the FLBWE bit (register PFE0CN) to select block write mode.
Step 3. Set the PSWE bit (register PSCTL).
Step 4. Clear the PSEE bit (register PSCTL).
Step 5. Write the first key code to FLKEY: 0xA5.
Step 6. Write the second key code to FLKEY: 0xF1.
Step 7. Using the MOVX instruction, write the first data byte to the even block location (ending in 0b).
Step 8. Write the first key code to FLKEY: 0xA5.
Step 9. Write the second key code to FLKEY: 0xF1.
Step 10. Using the MOVX instruction, write the second data byte to the odd block location (ending in 1b).
Step 11. Clear the PSWE bit.
Step 12. Re-enable interrupts.

Steps 5–10 must be repeated for each block to be written.

SFR Definition 13.2. EMI0CF: External Memory Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	USBFAE	-	EMD2	EMD1	EMD0	EALE1	EALE0	00000011
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x85

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

Bit6: USBFAE: USB FIFO Access Enable.
0: USB FIFO RAM not available through MOVX instructions.
1: USB FIFO RAM available using MOVX instructions. The 1k of USB RAM will be mapped in XRAM space at addresses 0x0400 to 0x07FF. **The USB clock must be active and greater than or equal to twice the SYSCLK (USBCLK $\geq$ 2 x SYSCLK) to access this area with MOVX instructions.**

Bit5: Unused. Read = 0b. Write = don't care.

Bit4: EMD2: EMIF Multiplex Mode Select.
0: EMIF operates in multiplexed address/data mode.
1: EMIF operates in non-multiplexed mode (separate address and data pins).

Bits3–2: EMD1–0: EMIF Operating Mode Select.
These bits control the operating mode of the External Memory Interface.
00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space.
01: Split Mode without Bank Select: Accesses below the on-chip XRAM boundary are directed on-chip. Accesses above the on-chip XRAM boundary are directed off-chip. 8-bit off-chip MOVX operations use the current contents of the Address High port latches to resolve upper address byte. Note that in order to access off-chip space, EMI0CN must be set to a page that is not contained in the on-chip address space.
10: Split Mode with Bank Select: Accesses below the on-chip XRAM boundary are directed on-chip. Accesses above the on-chip XRAM boundary are directed off-chip. 8-bit off-chip MOVX operations use the contents of EMI0CN to determine the high-byte of the address.
11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.

Bits1–0: EALE1–0: ALE Pulse-Width Select Bits (only has effect when EMD2 = 0).
00: ALE high and ALE low pulse width = 1 SYSCLK cycle.
01: ALE high and ALE low pulse width = 2 SYSCLK cycles.
10: ALE high and ALE low pulse width = 3 SYSCLK cycles.
11: ALE high and ALE low pulse width = 4 SYSCLK cycles.

SFR Definition 14.4. OSCXCN: External Oscillator Control

R	R/W	R/W	R/W	R	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB1

Bit7: XTLVLD: Crystal Oscillator Valid Flag.
(Read only when XOSCMD = 11x.)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits6–4: XOSCMD2–0: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode.
011: External CMOS Clock Mode with divide by 2 stage.
100: RC Oscillator Mode.
101: Capacitor Oscillator Mode.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit3: RESERVED. Read = 0, Write = don't care.

Bits2–0: XFCN2–0: External Oscillator Frequency Control Bits.
000-111: See table below:

XFCN	Crystal (XOSCMD = 11x)	RC (XOSCMD = 10x)	C (XOSCMD = 10x)
000	f ≤ 32 kHz	f ≤ 25 kHz	K Factor = 0.87
001	32 kHz < f ≤ 84kHz	25 kHz < f ≤ 50 kHz	K Factor = 2.6
010	84 kHz < f ≤ 225 kHz	50 kHz < f ≤ 100 kHz	K Factor = 7.7
011	225 kHz < f ≤ 590 kHz	100 kHz < f ≤ 200 kHz	K Factor = 22
100	590 kHz < f ≤ 1.5 MHz	200 kHz < f ≤ 400 kHz	K Factor = 65
101	1.5 MHz < f ≤ 4 MHz	400 kHz < f ≤ 800 kHz	K Factor = 180
110	4 MHz < f ≤ 10 MHz	800 kHz < f ≤ 1.6 MHz	K Factor = 664
111	10 MHz < f ≤ 30 MHz	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590

CRYSTAL MODE (Circuit from Figure 14.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal or resonator frequency.

RC MODE (Circuit from Figure 14.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
$f = 1.23(10^3) / (R \times C)$, where
f = frequency of clock in MHz
C = capacitor value in pF
R = Pull-up resistor value in kΩ

C MODE (Circuit from Figure 14.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
$f = KF / (C \times V_{DD})$, where
f = frequency of clock in MHz
C = capacitor value the XTAL2 pin in pF
V_{DD} = Power Supply on MCU in volts

15. Port Input/Output

Digital and analog resources are available through 40 I/O pins (48-pin packages) or 25 I/O pins (32-pin packages). Port pins are organized as shown in Figure 15.1. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input; Port pins P0.0-P3.7 can be assigned to one of the internal digital resources as shown in Figure 15.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the Priority Decoder (Figure 15.3 and Figure 15.4). The registers XBR0, XBR1, and XBR2 defined in SFR Definition 15.1, SFR Definition 15.2, and SFR Definition 15.3, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 15.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1,2,3,4). Complete Electrical Specifications for Port I/O are given in Table 15.1 on page 158.

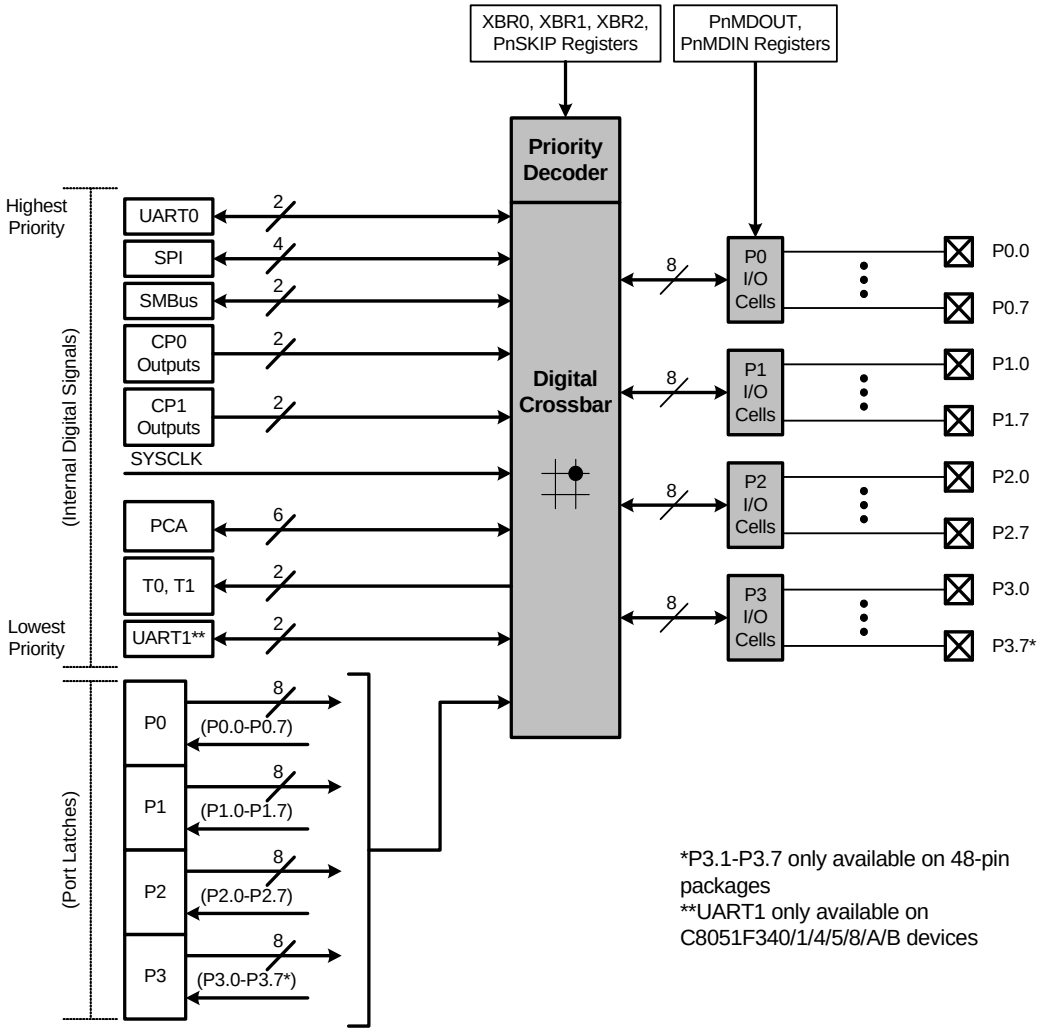

Figure 15.1. Port I/O Functional Block Diagram (Port 0 through Port 3)

USB Register Definition 16.8. POWER: USB0 Power

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
ISOUD	-	-	USBINH	USBRST	RESUME	SUSMD	SUSEN	00010000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x01

Bit7: ISOUD: ISO Update
This bit affects all IN Isochronous endpoints.
0: When software writes INPRDY = ‘1’, USB0 will send the packet when the next IN token is received.
1: When software writes INPRDY = ‘1’, USB0 will wait for a SOF token before sending the packet. If an IN token is received before a SOF token, USB0 will send a zero-length data packet.

Bits6–5: Unused. Read = 00b. Write = don’t care.

Bit4: USBINH: USB0 Inhibit
This bit is set to ‘1’ following a power-on reset (POR) or an asynchronous USB0 reset (see Bit3: RESET). Software should clear this bit after all USB0 and transceiver initialization is complete. Software cannot set this bit to ‘1’.
0: USB0 enabled.
1: USB0 inhibited. All USB traffic is ignored.

Bit3: USBRST: Reset Detect
Writing ‘1’ to this bit forces an asynchronous USB0 reset. Reading this bit provides bus reset status information.
Read:
0: Reset signaling is not present on the bus.
1: Reset signaling detected on the bus.

Bit2: RESUME: Force Resume
Software can force resume signaling on the bus to wake USB0 from suspend mode. Writing a ‘1’ to this bit while in Suspend mode (SUSMD = ‘1’) forces USB0 to generate Resume signaling on the bus (a remote Wakeup event). Software should write RESUME = ‘0’ after 10 ms to15 ms to end the Resume signaling. An interrupt is generated, and hardware clears SUSMD, when software writes RESUME = ‘0’.

Bit1: SUSMD: Suspend Mode
Set to ‘1’ by hardware when USB0 enters suspend mode. Cleared by hardware when software writes RESUME = ‘0’ (following a remote wakeup) or reads the CMINT register after detection of Resume signaling on the bus.
0: USB0 not in suspend mode.
1: USB0 in suspend mode.

Bit0: SUSEN: Suspend Detection Enable
0: Suspend detection disabled. USB0 will ignore suspend signaling on the bus.
1: Suspend detection enabled. USB0 will enter suspend mode if it detects suspend signaling on the bus.

USB Register Definition 16.20. EINCSRH: USB0 IN Endpoint Control High Byte

R/W	R/W	R/W	R	R/W	R/W	R	R	Reset Value
DBIEN	ISO	DIRSEL	-	FCDT	SPLIT	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x12

Bit7: DBIEN: IN Endpoint Double-buffer Enable.
0: Double-buffering disabled for the selected IN endpoint.
1: Double-buffering enabled for the selected IN endpoint.

Bit6: ISO: Isochronous Transfer Enable.
This bit enables/disables isochronous transfers on the current endpoint.
0: Endpoint configured for bulk/interrupt transfers.
1: Endpoint configured for isochronous transfers.

Bit5: DIRSEL: Endpoint Direction Select.
This bit is valid only when the selected FIFO is not split (SPLIT = ‘0’).
0: Endpoint direction selected as OUT.
1: Endpoint direction selected as IN.

Bit4: Unused. Read = ‘0’. Write = don’t care.

Bit3: FCDT: Force Data Toggle.
0: Endpoint data toggle switches only when an ACK is received following a data packet transmission.
1: Endpoint data toggle forced to switch after every data packet is transmitted, regardless of ACK reception.

Bit2: SPLIT: FIFO Split Enable.
When SPLIT = ‘1’, the selected endpoint FIFO is split. The upper half of the selected FIFO is used by the IN endpoint; the lower half of the selected FIFO is used by the OUT endpoint.

Bits1–0: Unused. Read = 00b; Write = don’t care.

16.13. Controlling Endpoints1-3 OUT

Endpoints1-3 OUT are managed via USB registers EOUTCSRL and EOUTCSRH. All OUT endpoints can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing ‘1’ to the ISO bit in register EOUTCSRH. Bulk and Interrupt transfers are handled identically by hardware.

An Endpoint1-3 OUT interrupt may be generated by the following:

1. Hardware sets the OPRDY bit (EINCSRL.0) to ‘1’.
2. Hardware generates a STALL condition.

16.13.1.Endpoints1-3 OUT Interrupt or Bulk Mode

When the ISO bit (EOUTCSRH.6) = ‘0’ the target endpoint operates in Bulk or Interrupt mode. Once an endpoint has been configured to operate in Bulk/Interrupt OUT mode (typically following an Endpoint0 SET_INTERFACE command), hardware will set the OPRDY bit (EOUTCSRL.0) to ‘1’ and generate an interrupt upon reception of an OUT token and data packet. The number of bytes in the current OUT data packet (the packet ready to be unloaded from the FIFO) is given in the EOUTCNTH and EOUTCNTL registers. In response to this interrupt, firmware should unload the data packet from the OUT FIFO and reset the OPRDY bit to ‘0’.

USB Register Definition 16.22. EOUTCSRH: USB0 OUT Endpoint Control High Byte

R/W	R/W	R/W	R/W	R	R	R	R	Reset Value
DBOEN	ISO	-	-	-	-	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x15

Bit7: DBOEN: Double-buffer Enable
0: Double-buffering disabled for the selected OUT endpoint.
1: Double-buffering enabled for the selected OUT endpoint.
Bit6: ISO: Isochronous Transfer Enable
This bit enables/disables isochronous transfers on the current endpoint.
0: Endpoint configured for bulk/interrupt transfers.
1: Endpoint configured for isochronous transfers.
Bits5–0: Unused. Read = 000000b; Write = don't care.

USB Register Definition 16.23. EOUTCNTL: USB0 OUT Endpoint Count Low

R	R	R	R	R	R	R	R	Reset Value
EOCL								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x16

Bits7–0: EOCL: OUT Endpoint Count Low Byte
EOCL holds the lower 8-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = '1'.

USB Register Definition 16.24. EOUTCNTH: USB0 OUT Endpoint Count High

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	-	-	E0CH		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x17

Bits7–2: Unused. Read = 00000. Write = don't care.
Bits1–0: EOCH: OUT Endpoint Count High Byte
EOCH holds the upper 2-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = '1'.

17.5.2. Master Receiver Mode

Serial data is received 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 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. After each byte is received, ACKRQ is set to ‘1’ and an interrupt is generated. Software must write the ACK bit (SMB0CN.1) to define the outgoing acknowledge value (Note: writing a ‘1’ to the ACK bit generates an ACK; writing a ‘0’ generates a NACK). Software should write a ‘0’ to the ACK bit after the last byte is received, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. Note that the interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 17.6 shows a typical Master 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.

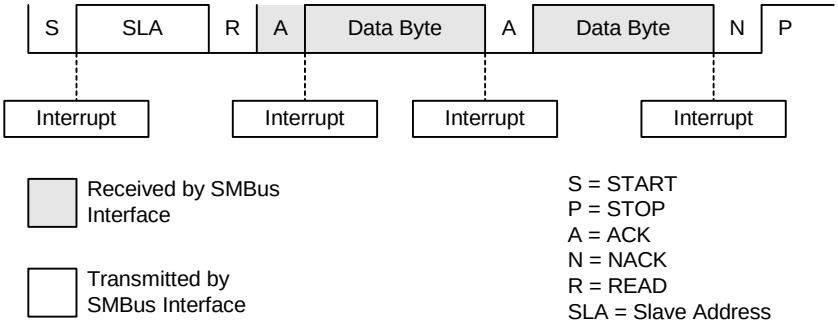

Figure 17.6. Typical Master Receiver Sequence

20.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

20.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

20.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

20.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

20.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 20.2, Figure 20.3, and Figure 20.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section "**15. Port Input/Output**" on page **142** for general purpose port I/O and crossbar information.

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.

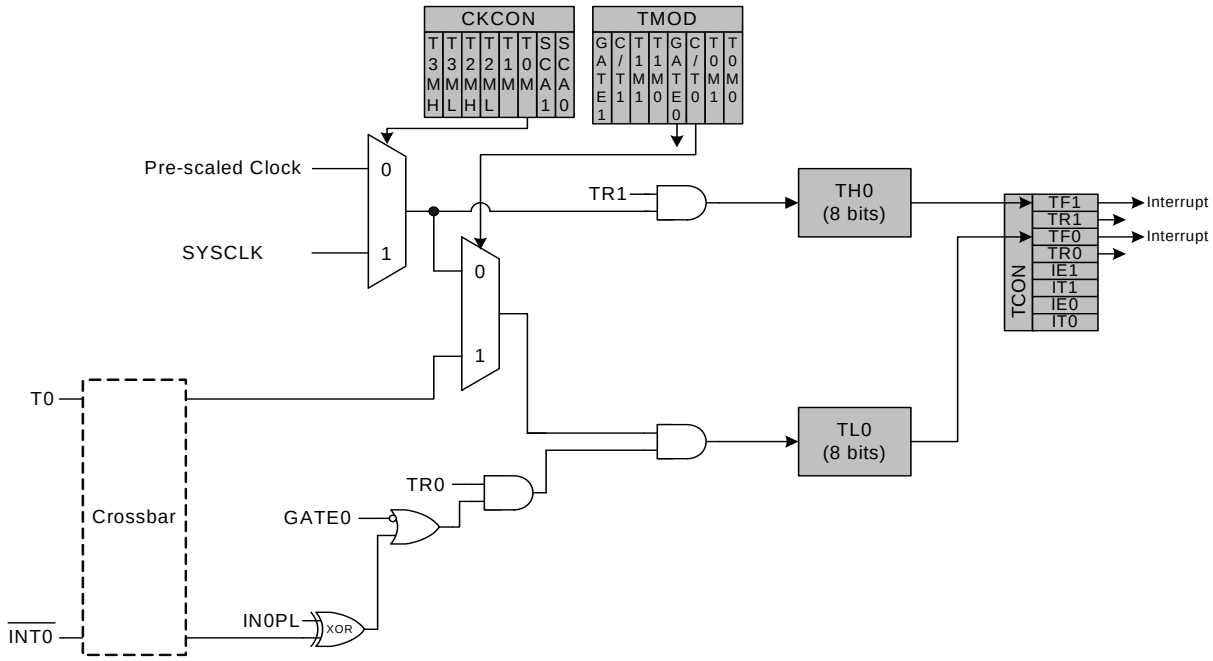

Figure 21.3. T0 Mode 3 Block Diagram

21.2. 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, USB Start-of-Frame (SOF) capture 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.

21.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT = ‘0’ and T2CE = ‘0’, 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 21.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled, 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.

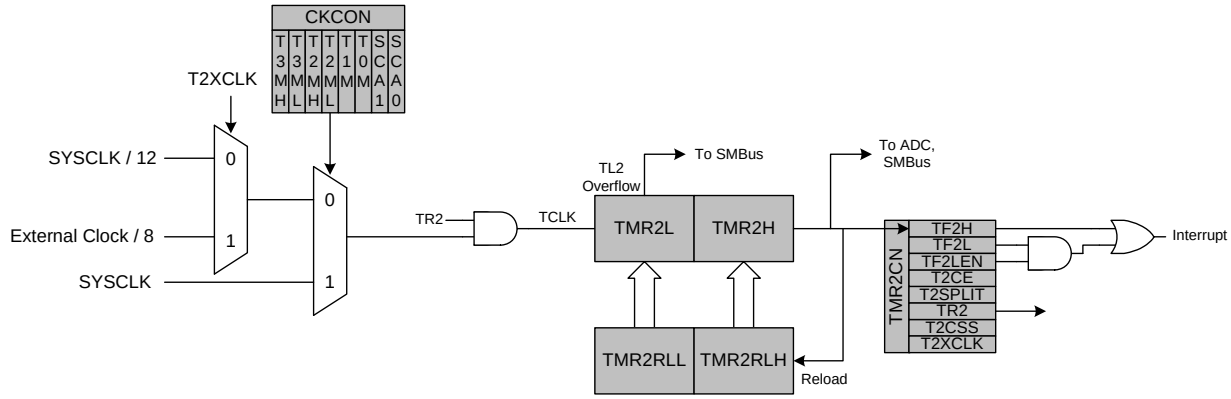

Figure 21.4. Timer 2 16-Bit Mode Block Diagram

22.2.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

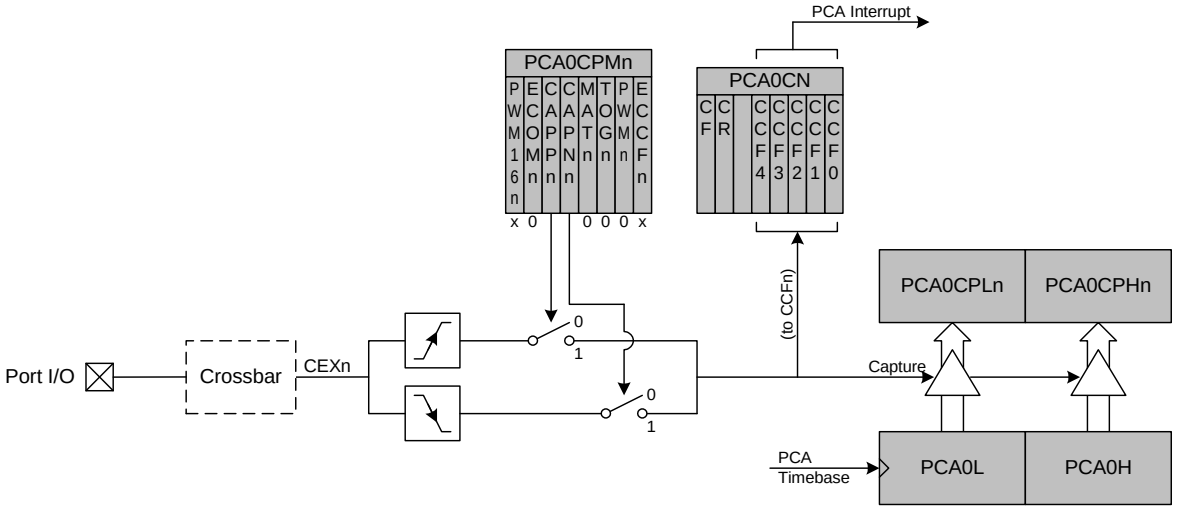

Figure 22.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.