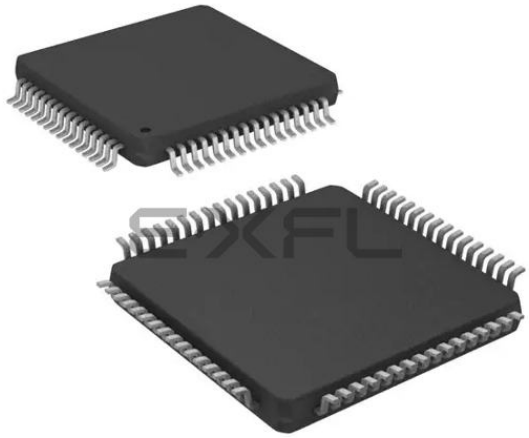


Welcome to E-XFL.COM

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	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	49
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66j50-i-pt

TABLE 1-4: PIC18F8XJ5X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
				PORTA is a bidirectional I/O port.
RA0/AN0	30			
RA0		I/O	TTL	Digital I/O.
AN0		I	Analog	Analog input 0.
RA1/AN1	29			
RA1		I/O	TTL	Digital I/O.
AN1		I	Analog	Analog input 1.
RA2/AN2/VREF-	28			
RA2		I/O	TTL	Digital I/O.
AN2		I	Analog	Analog input 2.
VREF-		I	Analog	A/D reference voltage (low) input.
RA3/AN3/VREF+	27			
RA3		I/O	TTL	Digital I/O.
AN3		I	Analog	Analog input 3.
VREF+		I	Analog	A/D reference voltage (high) input.
RA4/PMD5/T0CKI	34			
RA4		I/O	ST	Digital I/O.
PMD5(7)		I/O	TTL	Parallel Master Port data.
T0CKI		I	ST	Timer0 external clock input.
RA5/PMD4/AN4/C2INA	33			
RA5		I/O	TTL	Digital I/O.
PMD4(7)		I/O	TTL	Parallel Master Port data.
AN4		I	Analog	Analog input 4.
C2INA		I	Analog	Comparator 2 input A.
RA6	—	—	—	See the OSC2/CLKO/RA6 pin.
RA7	—	—	—	See the OSC1/CLKI/RA7 pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Pin placement when PMPMX = 1.

7: Pin placement when PMPMX = 0.

8: RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

TABLE 1-4: PIC18F8XJ5X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
				PORTD is a bidirectional I/O port.
RD0/AD0/PMD0	72			
RD0		I/O	ST	Digital I/O.
AD0		I/O	TTL	External memory address/data 0.
PMD0(6)		I/O	TTL	Parallel Master Port data.
RD1/AD1/PMD1	69			
RD1		I/O	ST	Digital I/O.
AD1		I/O	TTL	External memory address/data 1.
PMD1(6)		I/O	TTL	Parallel Master Port data.
RD2/AD2/PMD2	68			
RD2		I/O	ST	Digital I/O.
AD2		I/O	TTL	External memory address/data 2.
PMD2(6)		I/O	TTL	Parallel Master Port data.
RD3/AD3/PMD3	67			
RD3		I/O	ST	Digital I/O.
AD3		I/O	TTL	External memory address/data 3.
PMD3(6)		I/O	TTL	Parallel Master Port data.
RD4/AD4/PMD4/SDO2	66			
RD4		I/O	ST	Digital I/O.
AD4		I/O	TTL	External memory address/data 4.
PMD4(6)		I/O	TTL	Parallel Master Port data.
SDO2		O	—	SPI data out.
RD5/AD5/PMD5/SDI2/SDA2	65			
RD5		I/O	ST	Digital I/O.
AD5		I/O	TTL	External memory address/data 5.
PMD5(6)		I/O	TTL	Parallel Master Port data.
SDI2		I	ST	SPI data in.
SDA2		I/O	ST	I^2C™ data I/O.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Pin placement when PMPMX = 1.

7: Pin placement when PMPMX = 0.

8: RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

© 2009 Microchip Technology Inc.

5.0 MEMORY ORGANIZATION

There are two types of memory in PIC18 Flash microcontroller devices:

- Program Memory
- Data RAM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces.

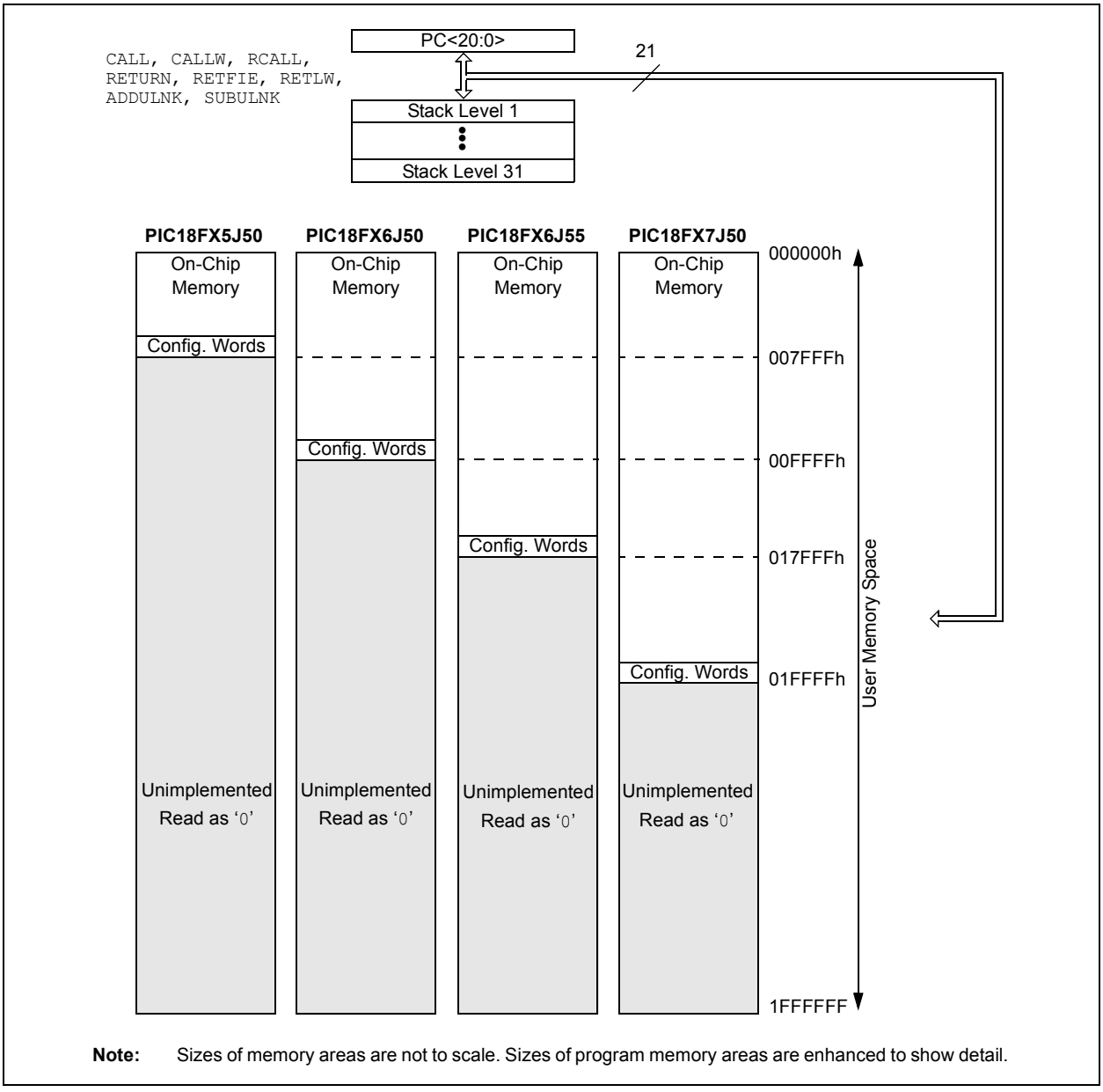
Additional detailed information on the operation of the Flash program memory is provided in **Section 6.0 "Flash Program Memory"**.

5.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The entire PIC18F87J10 family offers a range of on-chip Flash program memory sizes, from 64 Kbytes (up to 16,384 single-word instructions) to 128 Kbytes (65,536 single-word instructions). The program memory maps for individual family members are shown in Figure 5-3.

FIGURE 5-1: MEMORY MAPS FOR PIC18F87J50 FAMILY DEVICES



Note: Sizes of memory areas are not to scale. Sizes of program memory areas are enhanced to show detail.

6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 64 bytes at a time or two bytes at a time. Program memory is erased in blocks of 1024 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

6.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

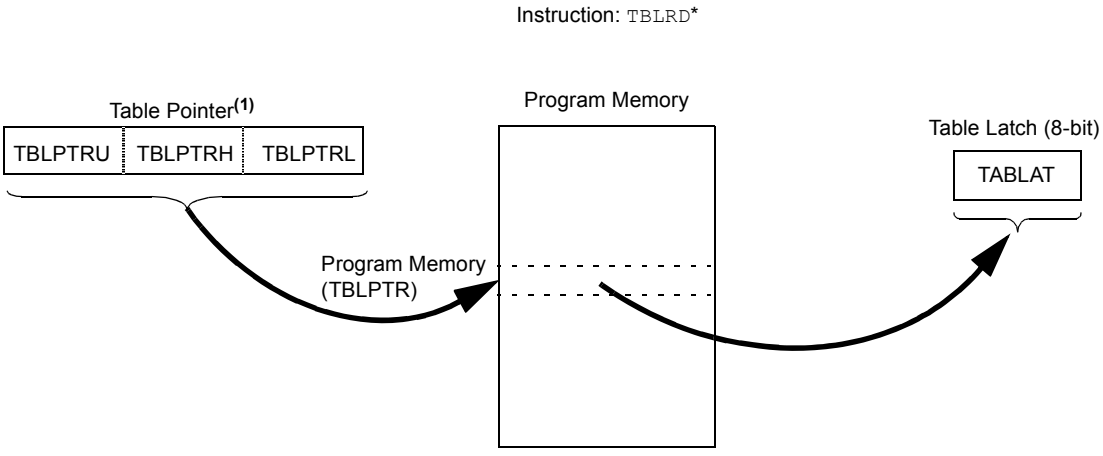
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and place it into the data RAM space. Figure 6-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 6.5 "Writing to Flash Program Memory"**. Figure 6-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word-aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned.

FIGURE 6-1: TABLE READ OPERATION



Note 1: Table Pointer register points to a byte in program memory.

TABLE 10-6: PORTB FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RB0/FLT0/INT0	RB0	0	O	DIG	LATB<0> data output.
		1	I	TTL	PORTB<0> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	FLT0	1	I	ST	Enhanced PWM Fault input (ECCP1 module); enabled in software.
	INT0	1	I	ST	External interrupt 0 input.
RB1/INT1/ PMA4	RB1	0	O	DIG	LATB<1> data output.
		1	I	TTL	PORTB<1> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	INT1	1	I	ST	External interrupt 1 input.
	PMA4	x	O	—	Parallel Master Port address out.
RB2/INT2/ PMA3	RB2	0	O	DIG	LATB<2> data output.
		1	I	TTL	PORTB<2> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	INT2	1	I	ST	External interrupt 2 input.
	PMA3	x	O	—	Parallel Master Port address out.
RB3/INT3/ ECCP2/P2A/ PMA2	RB3	0	O	DIG	LATB<3> data output.
		1	I	TTL	PORTB<3> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	INT3	1	I	ST	External interrupt 3 input.
	ECCP2(1)	0	O	DIG	ECCP2 compare output and ECCP2 PWM output; takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A(1)	0	O	DIG	ECCP2 Enhanced PWM output, channel A. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
	PMA2	x	O	—	Parallel Master Port address out.
RB4/KBI0/ PMA1	RB4	0	O	DIG	LATB<4> data output.
		1	I	TTL	PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI0		I	TTL	Interrupt-on-pin change.
	PMA1	x	O	—	Parallel Master Port address out.
RB5/KBI1/ PMA0	RB5	0	O	DIG	LATB<5> data output.
		1	I	TTL	PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI1		I	TTL	Interrupt-on-pin change.
	PMA0	x	O	—	Parallel Master Port address out.
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI2	1	I	TTL	Interrupt-on-pin change.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation.(2)
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI3	1	I	TTL	Interrupt-on-pin change.
	PGD	x	O	DIG	Serial execution data output for ICSP and ICD operation.(2)
		x	I	ST	Serial execution data input for ICSP and ICD operation.(2)

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate assignment for ECCP2/P2A when the CCP2MX Configuration bit is cleared (Extended Microcontroller mode, 80-pin devices only). Default assignment is RC1.

2: All other pin functions are disabled when ICSP™ or ICD are enabled.

© 2009 Microchip Technology Inc.

The CCPRxH register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2 (TMR4), concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 (TMR4) prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by Equation 17-3:

EQUATION 17-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCPx pin will not be cleared.

17.4.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 (PR4) register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 (TMR4) prescale value, then enable Timer2 (Timer4) by writing to T2CON (T4CON).
5. Configure the CCPx module for PWM operation.

TABLE 17-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

19.3.4 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPxCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPxCON registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and $\overline{SS}$x pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDIx is automatically controlled by the SPI module
- SDOx must have the TRISC<5> or TRISD<4> bit cleared
- SCKx (Master mode) must have the TRISC<3> or TRISD<6>bit cleared
- SCKx (Slave mode) must have the TRISC<3> or TRISD<6> bit set
- $\overline{SS}$x must have the TRISF<7> or TRISD<7> bit set

Any serial port function that is not desired may be overridden by programming the corresponding Data Direction (TRIS) register to the opposite value.

19.3.5 TYPICAL CONNECTION

Figure 19-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 19-2: SPI MASTER/SLAVE CONNECTION

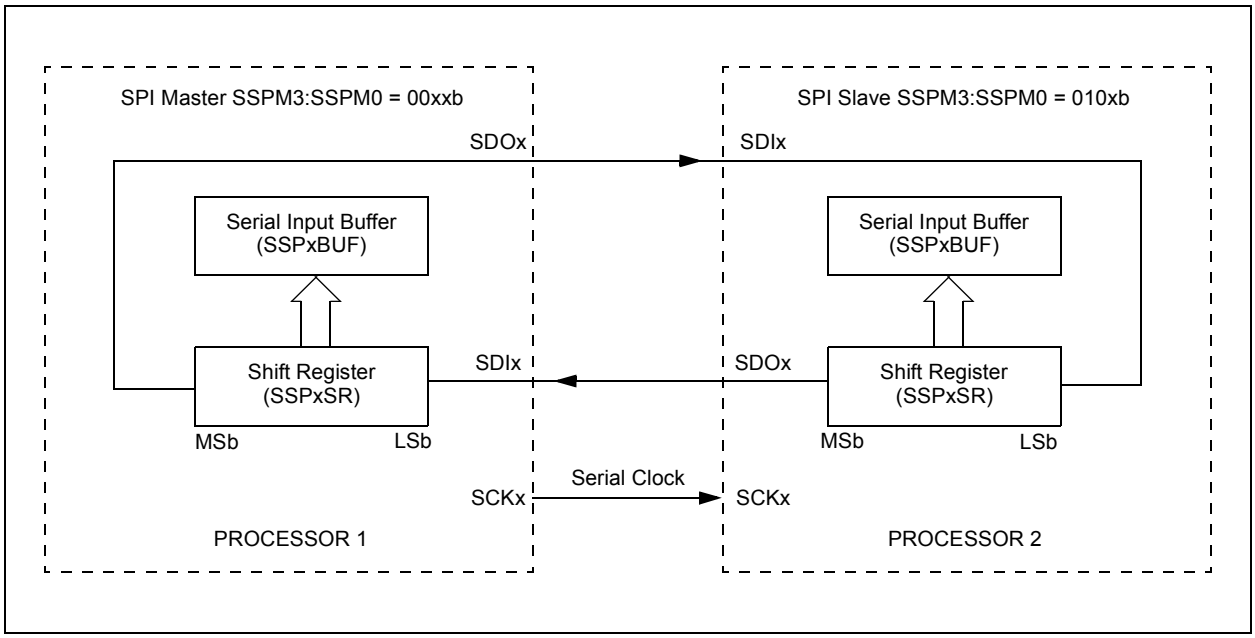
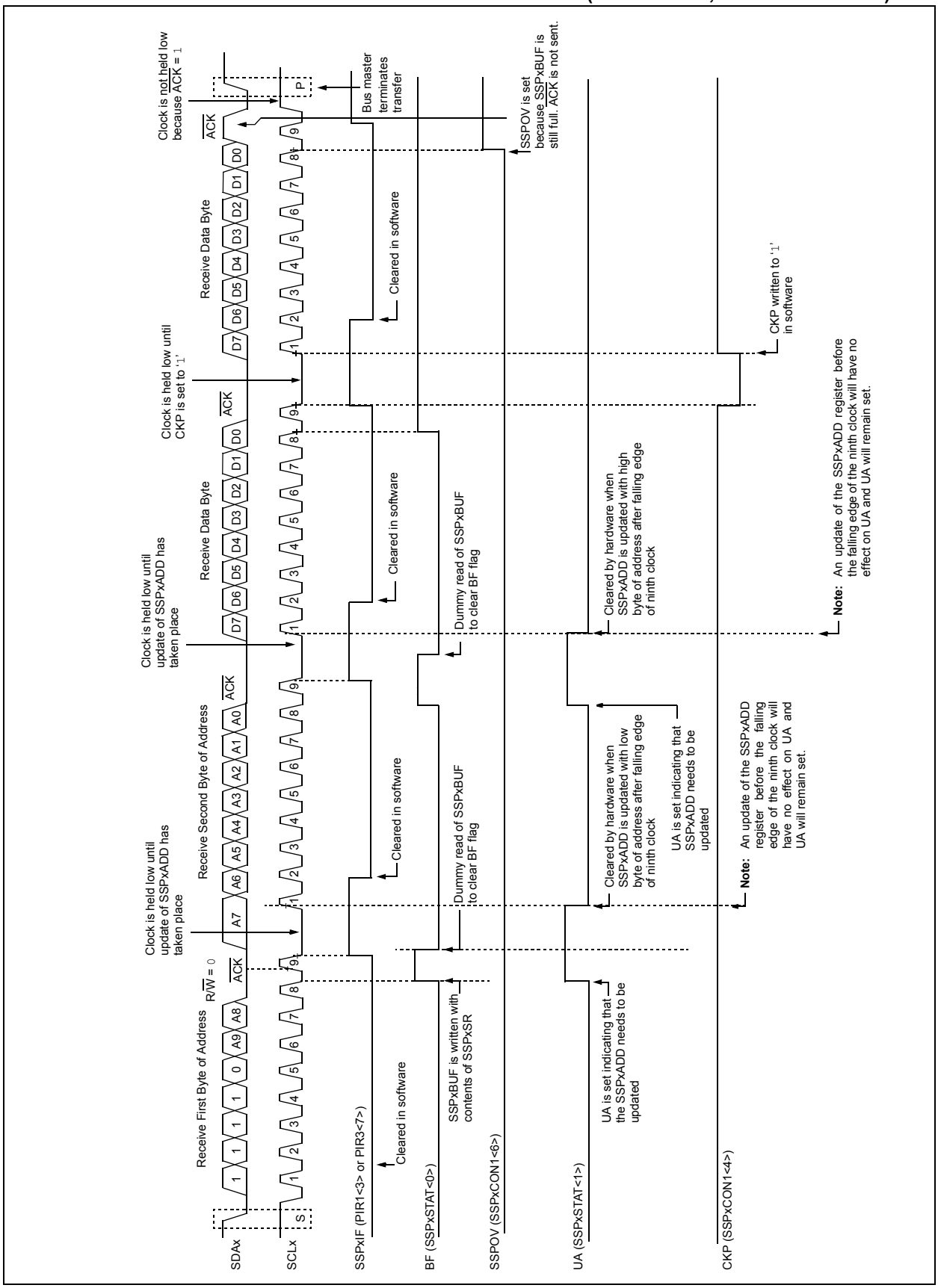


FIGURE 19-16: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)



REGISTER 20-1: TXSTAx: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don't care.

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-Bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)

1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit

Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed

Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit

1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

© 2009 Microchip Technology Inc.

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9615.	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

20.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>) are set while the Transmit Shift Register is loaded with data. Note that the value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

Note that the data value written to the TXREGx for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 20-10 for the timing of the Break character sequence.

20.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

20.2.6 RECEIVING A BREAK CHARACTER

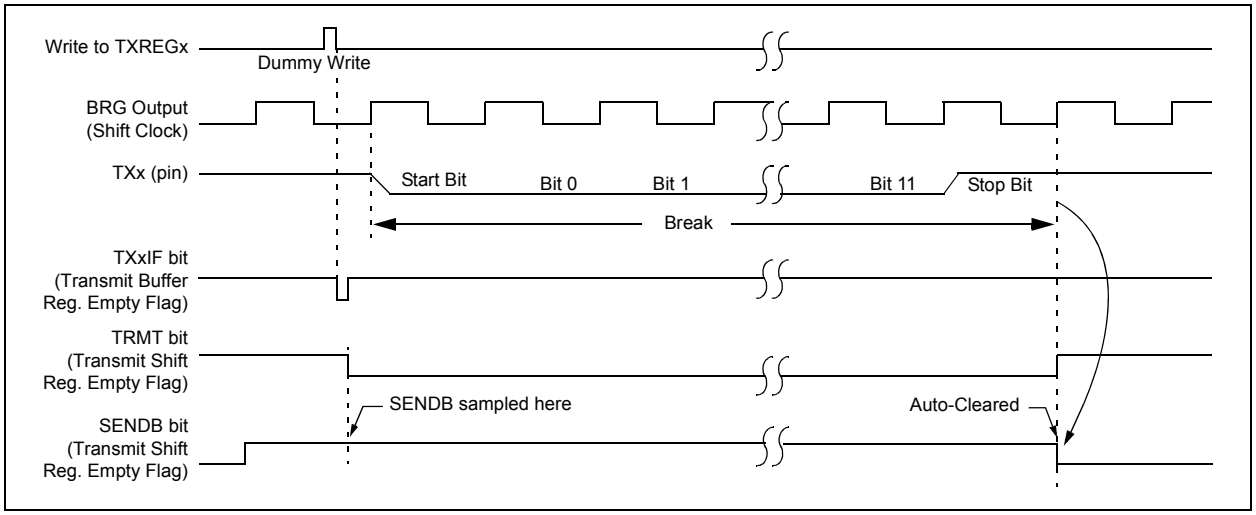
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 20.2.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXxIF interrupt is observed.

FIGURE 20-10: SEND BREAK CHARACTER SEQUENCE



© 2009 Microchip Technology Inc.

22.2 USB Status and Control

The operation of the USB module is configured and managed through three control registers. In addition, a total of 22 registers are used to manage the actual USB transactions. The registers are:

- USB Control register (UCON)
- USB Configuration register (UCFG)
- USB Transfer Status register (USTAT)
- USB Device Address register (UADDR)
- Frame Number registers (UFRMH:UFRML)
- Endpoint Enable registers 0 through 15 (UEPn)

22.2.1 USB CONTROL REGISTER (UCON)

The USB Control register (Register 22-1) contains bits needed to control the module behavior during transfers. The register contains bits that control the following:

- Main USB Peripheral Enable
- Ping-Pong Buffer Pointer Reset
- Control of the Suspend mode
- Packet Transfer Disable

In addition, the USB Control register contains a status bit, SE0 (UCON<5>), which is used to indicate the occurrence of a single-ended zero on the bus. When the USB module is enabled, this bit should be monitored to determine whether the differential data lines have come out of a single-ended zero condition. This helps to differentiate the initial power-up state from the USB Reset signal.

The overall operation of the USB module is controlled by the USBEN bit (UCON<3>). Setting this bit activates the module and resets all of the PPBI bits in the Buffer Descriptor Table to '0'. This bit also activates the internal pull-up resistors, if they are enabled. Thus, this bit can be used as a soft attach/detach to the USB. Although all status and control bits are ignored when this bit is clear, the module needs to be fully preconfigured prior to setting this bit. This bit cannot be set until the USB module is supplied with an active clock source. If the PLL is being used, it should be enabled at least two milliseconds (enough time for the PLL to lock) before attempting to set the USBEN bit.

REGISTER 22-1: UCON: USB CONTROL REGISTER

U-0	R/W-0	R-x	R/C-0	R/W-0	R/W-0	R/W-0	U-0
—	PPBRST	SE0	PKTDIS	USBEN(1)	RESUME	SUSPND	—
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **PPBRST:** Ping-Pong Buffers Reset bit
- 1 = Reset all Ping-Pong Buffer Pointers to the Even Buffer Descriptor (BD) banks
- 0 = Ping-Pong Buffer Pointers not being reset

bit 5 **SE0:** Live Single-Ended Zero Flag bit
- 1 = Single-ended zero active on the USB bus
- 0 = No single-ended zero detected

bit 4 **PKTDIS:** Packet Transfer Disable bit
- 1 = SIE token and packet processing disabled, automatically set when a SETUP token is received
- 0 = SIE token and packet processing enabled

bit 3 **USBEN:** USB Module Enable bit(1)
- 1 = USB module and supporting circuitry enabled (device attached)
- 0 = USB module and supporting circuitry disabled (device detached)

bit 2 **RESUME:** Resume Signaling Enable bit
- 1 = Resume signaling activated
- 0 = Resume signaling disabled

bit 1 **SUSPND:** Suspend USB bit
- 1 = USB module and supporting circuitry in Power Conserve mode, SIE clock inactive
- 0 = USB module and supporting circuitry in normal operation, SIE clock clocked at the configured rate

bit 0 **Unimplemented**: Read as '0'

Note 1: This bit cannot be set if the USB module does not have an appropriate clock source.

© 2009 Microchip Technology Inc.

22.2.4 USB ENDPOINT CONTROL

Each of the 16 possible bidirectional endpoints has its own independent control register, UEPn (where 'n' represents the endpoint number). Each register has an identical complement of control bits. The prototype is shown in Register 22-4.

The EPHSHK bit (UEPn<4>) controls handshaking for the endpoint; setting this bit enables USB handshaking. Typically, this bit is always set except when using isochronous endpoints.

The EPCONDIS bit (UEPn<3>) is used to enable or disable USB control operations (SETUP) through the endpoint. Clearing this bit enables SETUP transactions. Note that the corresponding EPINEN and EPOUTEN bits must be set to enable IN and OUT transactions. For Endpoint 0, this bit should always be cleared since the USB specifications identify Endpoint 0 as the default control endpoint.

The EPOUTEN bit (UEPn<2>) is used to enable or disable USB OUT transactions from the host. Setting this bit enables OUT transactions. Similarly, the EPINEN bit (UEPn<1>) enables or disables USB IN transactions from the host.

The EPSTALL bit (UEPn<0>) is used to indicate a STALL condition for the endpoint. If a STALL is issued on a particular endpoint, the EPSTALL bit for that endpoint pair will be set by the SIE. This bit remains set until it is cleared through firmware, or until the SIE is reset.

REGISTER 22-4: UEPn: USB ENDPOINT n CONTROL REGISTER (UEP0 THROUGH UEP15)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EPHSHK:** Endpoint Handshake Enable bit
1 = Endpoint handshake enabled
0 = Endpoint handshake disabled (typically used for isochronous endpoints)

bit 3 **EPCONDIS:** Bidirectional Endpoint Control bit
If EPOUTEN = 1 and EPINEN = 1:
1 = Disable Endpoint n from control transfers; only IN and OUT transfers allowed
0 = Enable Endpoint n for control (SETUP) transfers; IN and OUT transfers also allowed

bit 2 **EPOUTEN:** Endpoint Output Enable bit
1 = Endpoint n output enabled
0 = Endpoint n output disabled

bit 1 **EPINEN:** Endpoint Input Enable bit
1 = Endpoint n input enabled
0 = Endpoint n input disabled

bit 0 **EPSTALL:** Endpoint Stall Enable bit(1)
1 = Endpoint n is stalled
0 = Endpoint n is not stalled

Note 1: Valid only if Endpoint n is enabled; otherwise, the bit is ignored.

22.9 Overview of USB

This section presents some of the basic USB concepts and useful information necessary to design a USB device. Although much information is provided in this section, there is a plethora of information provided within the USB specifications and class specifications. Thus, the reader is encouraged to refer to the USB specifications for more information (www.usb.org). If you are very familiar with the details of USB, then this section serves as a basic, high-level refresher of USB.

22.9.1 LAYERED FRAMEWORK

USB device functionality is structured into a layered framework graphically shown in Figure 22-12. Each level is associated with a functional level within the device. The highest layer, other than the device, is the configuration. A device may have multiple configurations. For example, a particular device may have multiple power requirements based on Self-Power Only or Bus Power Only modes.

For each configuration, there may be multiple interfaces. Each interface could support a particular mode of that configuration.

Below the interface is the endpoint(s). Data is directly moved at this level. There can be as many as 16 bidirectional endpoints. Endpoint 0 is always a control endpoint and by default, when the device is on the bus, Endpoint 0 must be available to configure the device.

22.9.2 FRAMES

Information communicated on the bus is grouped into 1 ms time slots, referred to as frames. Each frame can contain many transactions to various devices and endpoints. Figure 22-8 shows an example of a transaction within a frame.

22.9.3 TRANSFERS

There are four transfer types defined in the USB specification.

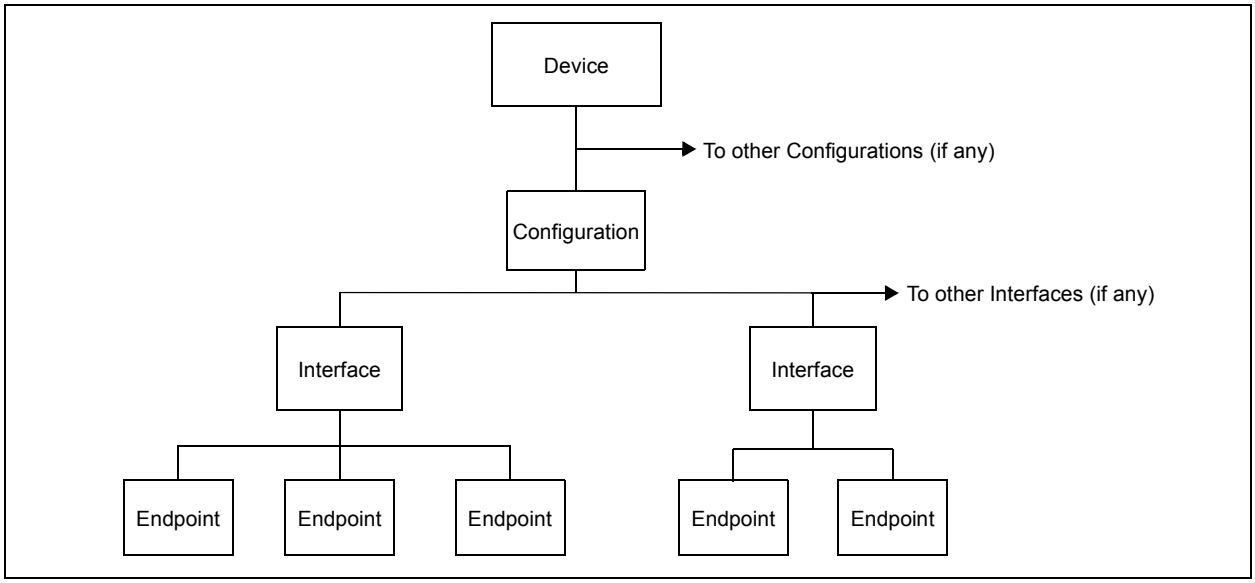
- **Isochronous:** This type provides a transfer method for large amounts of data (up to 1023 bytes) with timely delivery ensured; however, the data integrity is not ensured. This is good for streaming applications where small data loss is not critical, such as audio.
- **Bulk:** This type of transfer method allows for large amounts of data to be transferred with ensured data integrity; however, the delivery timeliness is not ensured.
- **Interrupt:** This type of transfer provides for ensured timely delivery for small blocks of data, plus data integrity is ensured.
- **Control:** This type provides for device setup control.

While full-speed devices support all transfer types, low-speed devices are limited to interrupt and control transfers only.

22.9.4 POWER

Power is available from the Universal Serial Bus. The USB specification defines the bus power requirements. Devices may either be self-powered or bus powered. Self-powered devices draw power from an external source, while bus powered devices use power supplied from the bus.

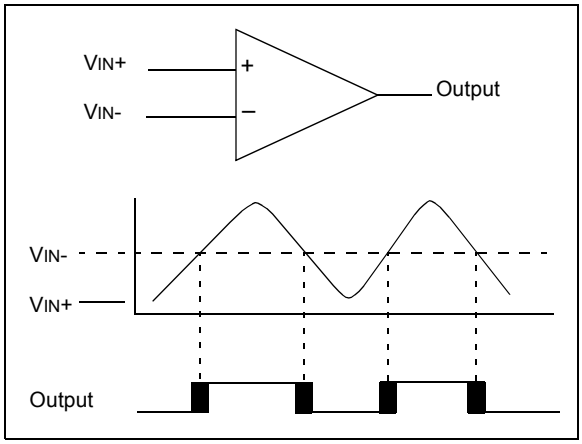
FIGURE 22-12: USB LAYERS



23.2 Comparator Operation

A single comparator is shown in Figure 23-2, along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 23-2 represent the uncertainty due to input offsets and response time.

FIGURE 23-2: SINGLE COMPARATOR



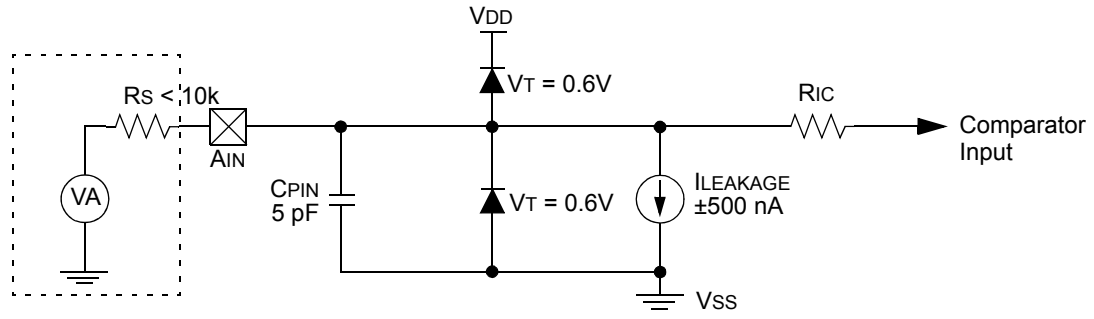
23.3 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response to a comparator input change. Otherwise, the maximum delay of the comparators should be used (see **Section 28.0 "Electrical Characteristics"**).

23.4 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 23-3. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

FIGURE 23-3: COMPARATOR ANALOG INPUT MODEL



Legend:

CPIN	=	Input Capacitance
VT	=	Threshold Voltage
ILEAKAGE	=	Leakage Current at the pin due to various junctions
RIC	=	Interconnect Resistance
RS	=	Source Impedance
VA	=	Analog Voltage

© 2009 Microchip Technology Inc.

25.0 SPECIAL FEATURES OF THE CPU

PIC18F87J10 family devices include several features intended to maximize reliability and minimize cost through elimination of external components. These are:

- Oscillator Selection
- Resets:
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor
- Two-Speed Start-up
- Code Protection
- In-Circuit Serial Programming

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 2.0 "Oscillator Configurations"**.

A complete discussion of device Resets and interrupts is available in previous sections of this data sheet. In addition to their Power-up and Oscillator Start-up Timers provided for Resets, the PIC18F87J10 family of devices have a configurable Watchdog Timer which is controlled in software.

The inclusion of an internal RC oscillator also provides the additional benefits of a Fail-Safe Clock Monitor (FSCM) and Two-Speed Start-up. FSCM provides for background monitoring of the peripheral clock and automatic switchover in the event of its failure. Two-Speed Start-up enables code to be executed almost immediately on start-up, while the primary clock source completes its start-up delays.

All of these features are enabled and configured by setting the appropriate Configuration register bits.

25.1 Configuration Bits

The Configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped starting at program memory location 300000h. A complete list is shown in Table 25-2. A detailed explanation of the various bit functions is provided in Register 25-1 through Register 25-6.

25.1.1 CONSIDERATIONS FOR CONFIGURING THE PIC18F87J50 FAMILY DEVICES

Unlike previous PIC18 microcontrollers, devices of the PIC18F87J10 family do not use persistent memory registers to store configuration information. The configuration bytes are implemented as volatile memory, which means that configuration data must be programmed each time the device is powered up.

Configuration data is stored in the four words at the top of the on-chip program memory space, known as the Flash Configuration Words. It is stored in program memory in the same order shown in Table 25-2, with CONFIG1L at the lowest address and CONFIG3H at the highest. The data is automatically loaded in the proper Configuration registers during device power-up.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The volatile memory cells used for the Configuration bits always reset to '1' on Power-on Resets. For all other type of Reset events, the previously programmed values are maintained and used without reloading from program memory.

The four Most Significant bits of CONFIG1H, CONFIG2H and CONFIG3H in program memory should also be '1111'. This makes these Configuration Words appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing '1's to these locations has no effect on device operation.

To prevent inadvertent configuration changes during code execution, all programmable Configuration bits are write-once. After a bit is initially programmed during a power cycle, it cannot be written to again. Changing a device configuration requires that power to the device be cycled.

26.0 INSTRUCTION SET SUMMARY

The PIC18F87J10 family of devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

26.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® instruction sets, while maintaining an easy migration from these PIC instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 26-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 26-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator, ‘f’, specifies which file register is to be used by the instruction. The destination designator, ‘d’, specifies where the result of the operation is to be placed. If ‘d’ is ‘0’, the result is placed in the WREG register. If ‘d’ is ‘1’, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator ‘b’ selects the number of the bit affected by the operation, while the file register designator, ‘f’, represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The **control** instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the CALL or RETURN instructions (specified by ‘s’)
- The mode of the table read and table write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 26-1 shows the general formats that the instructions can have. All examples use the convention ‘nnh’ to represent a hexadecimal number.

The instruction set summary, shown in Table 26-2, lists the standard instructions recognized by the Microchip MPASM™ Assembler.

Section 26.1.1 “Standard Instruction Set” provides a description of each instruction.

28.2 DC Characteristics: Power-Down and Supply Current PIC18F87J50 Family (Industrial) (Continued)

PIC18F87J50 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (IDD)(2)						
	All devices	5	14.2	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 31 kHz (**RC_RUN** mode, internal oscillator source)
		5.5	14.2	µA	+25°C		
		10	19.0	µA	+85°C		
	All devices	6.8	16.5	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		7.6	16.5	µA	+25°C		
		14	22.4	µA	+85°C		
	All devices	37	84	µA	-40°C	VDD = 3.3V(5)	
		51	84	µA	+25°C		
		72	108	µA	+85°C		
	All devices	0.43	0.82	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (**RC_RUN** mode, internal oscillator source)
		0.47	0.82	mA	+25°C		
		0.52	0.95	mA	+85°C		
	All devices	0.52	0.98	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.57	0.98	mA	+25°C		
		0.63	1.10	mA	+85°C		
	All devices	0.59	0.96	mA	-40°C	VDD = 3.3V(5)	
		0.65	0.96	mA	+25°C		
		0.72	1.18	mA	+85°C		
	All devices	0.88	1.45	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (**RC_RUN** mode, internal oscillator source)
		1.0	1.45	mA	+25°C		
		1.1	1.58	mA	+85°C		
	All devices	1.2	1.72	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		1.3	1.72	mA	+25°C		
		1.4	1.85	mA	+85°C		
	All devices	1.3	2.87	mA	-40°C	VDD = 3.3V(5)	
		1.4	2.87	mA	+25°C		
		1.5	2.96	mA	+85°C		

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta current are disabled (USB module, WDT, etc.).
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT disabled unless otherwise specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD), REGSLP = 1.

6: This is the module differential current when the USB module is enabled and clocked at 48 MHz, but with no USB cable attached. When the USB cable is attached or data is being transmitted, the current consumption may be much higher (see **Section 22.6.4 "USB Transceiver Current Consumption"**). During USB Suspend mode (USBEN = 1, SUSPND = 1, bus in Idle state), the USB module current will be dominated by the D+ or D- pull-up resistor. The integrated pull-up resistors use "resistor switching" according to the resistor_ecn supplement to the USB 2.0 specifications, and therefore, may be as low as 900Ω during Idle conditions.

28.2 DC Characteristics: Power-Down and Supply Current PIC18F87J50 Family (Industrial) (Continued)

PIC18F87J50 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (IDD) Cont.(2)						
	All devices	3	9.4	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 31 kHz (**RC_IDLE** mode, internal oscillator source)
		3.3	9.4	µA	+25°C		
		8.5	17.2	µA	+85°C		
	All devices	4	10.5	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		4.3	10.5	µA	+25°C		
		10.3	19.5	µA	+85°C		
	All devices	34	82	µA	-40°C	VDD = 3.3V(5)	
		48	82	µA	+25°C		
		69	105	µA	+85°C		
	All devices	0.33	0.75	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (**RC_IDLE** mode, internal oscillator source)
		0.37	0.75	mA	+25°C		
		0.41	0.84	mA	+85°C		
	All devices	0.39	0.78	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.42	0.78	mA	+25°C		
		0.47	0.91	mA	+85°C		
	All devices	0.43	0.82	mA	-40°C	VDD = 3.3V(5)	
		0.48	0.82	mA	+25°C		
		0.54	0.95	mA	+85°C		
	All devices	0.53	0.98	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (**RC_IDLE** mode, internal oscillator source)
		0.57	0.98	mA	+25°C		
		0.61	1.12	mA	+85°C		
	All devices	0.63	1.14	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.67	1.14	mA	+25°C		
		0.72	1.25	mA	+85°C		
	All devices	0.70	1.27	mA	-40°C	VDD = 3.3V(5)	
		0.76	1.27	mA	+25°C		
		0.82	1.45	mA	+85°C		

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta current are disabled (USB module, WDT, etc.).
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT disabled unless otherwise specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD), REGSLP = 1.

6: This is the module differential current when the USB module is enabled and clocked at 48 MHz, but with no USB cable attached. When the USB cable is attached or data is being transmitted, the current consumption may be much higher (see **Section 22.6.4 "USB Transceiver Current Consumption"**). During USB Suspend mode (USBEN = 1, SUSPND = 1, bus in Idle state), the USB module current will be dominated by the D+ or D- pull-up resistor. The integrated pull-up resistors use "resistor switching" according to the resistor_ecn supplement to the USB 2.0 specifications, and therefore, may be as low as 900Ω during Idle conditions.

© 2009 Microchip Technology Inc.

28.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 28-4: EXTERNAL CLOCK TIMING

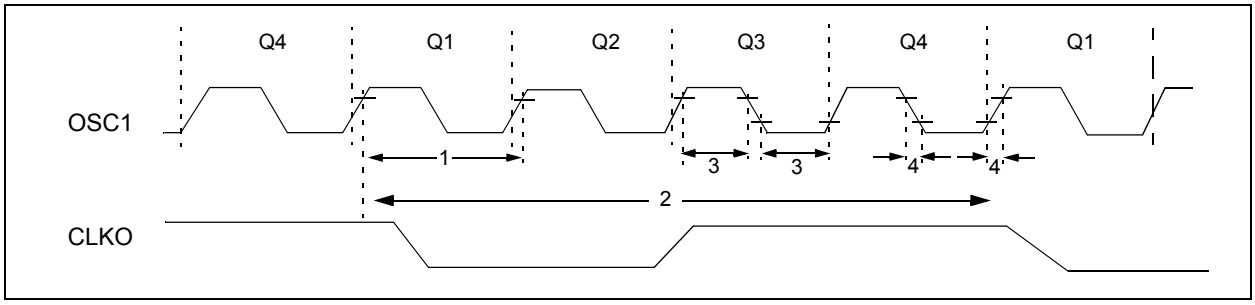


TABLE 28-7: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	FOSC	External CLKI Frequency(1)	DC	48	MHz	EC Oscillator mode
			DC	48		ECPLL Oscillator mode(2)
		Oscillator Frequency(1)	4	25	MHz	HS Oscillator mode
			4	25		HSPLL Oscillator mode(3)
1	TOSC	External CLKI Period(1)	20.8	—	ns	EC Oscillator mode
			20.8	—		ECPLL Oscillator mode(2)
		Oscillator Period(1)	40.0	250	ns	HS Oscillator mode
			40.0	250		HSPLL Oscillator mode(3)
2	TCY	Instruction Cycle Time(1)	83.3	—	ns	TCY = 4/FOSC, Industrial
3	TOSL, TOSH	External Clock in (OSC1) High or Low Time	10	—	ns	EC Oscillator mode
4	TOSR, TOSF	External Clock in (OSC1) Rise or Fall Time	—	7.5	ns	EC Oscillator mode

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.

2: In order to use the PLL, the external clock frequency must be either 4, 8, 12, 16, 20, 24, 40 or 48 MHz.

3: In order to use the PLL, the crystal/resonator must produce a frequency of either 4, 8, 12, 16, 20 or 24 MHz.