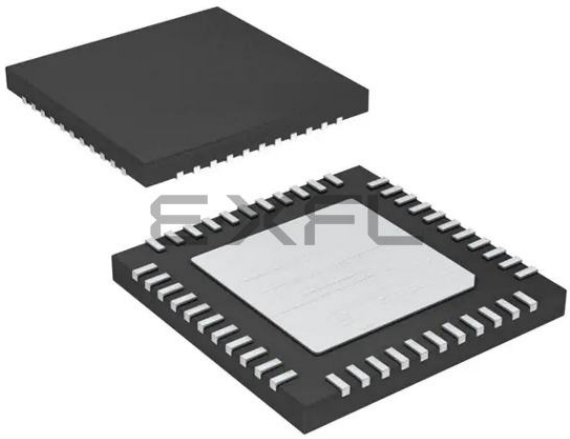
Microchip Technology - PIC18LF46J53-I/ML Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

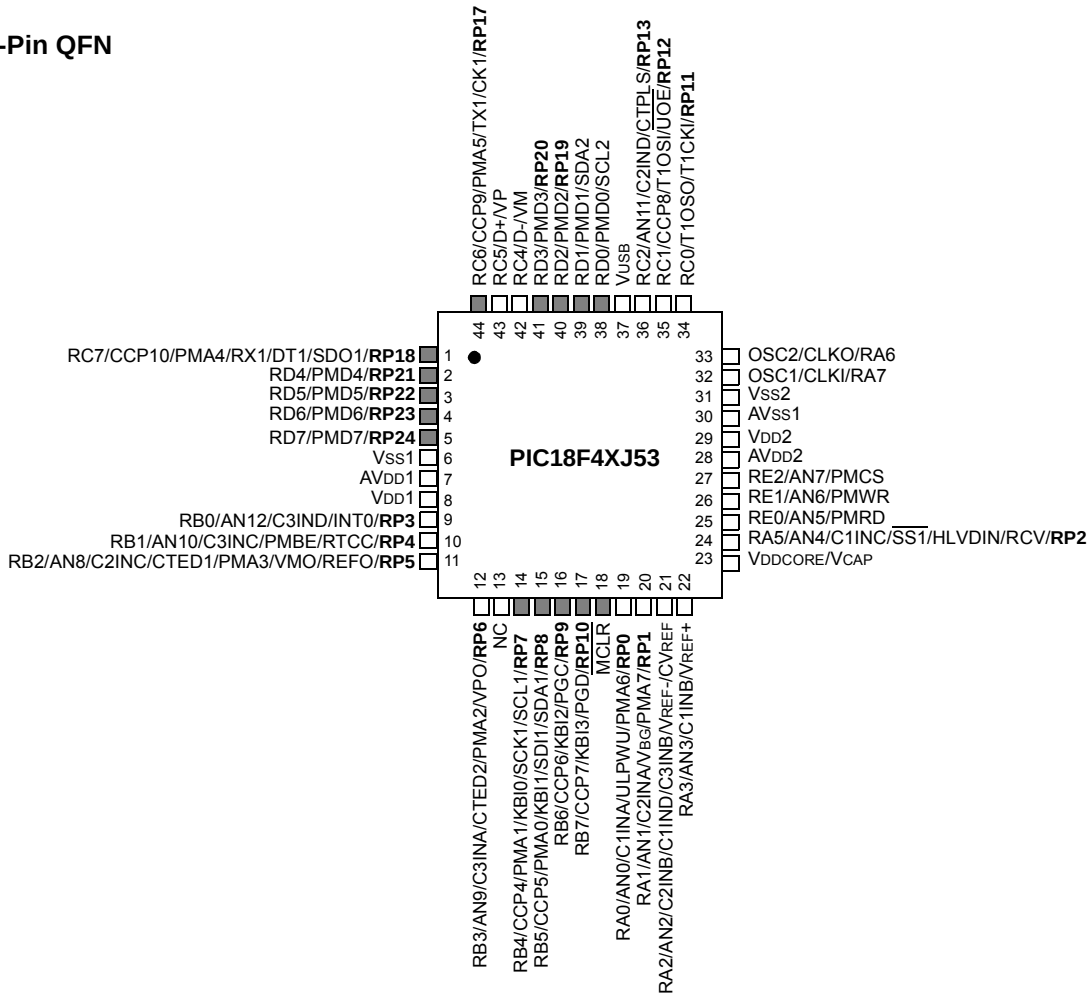
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 2.75V
Data Converters	A/D 13x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46j53-i-ml

Pin Diagrams (Continued)

44-Pin QFN



Legend: **RPn** represents remappable pins.
Shaded pins are 5.5V tolerant.

Note: For the QFN package, it is recommended that the bottom pad be connected to VSS.

REGISTER 3-1: OSCTUNE: OSCILLATOR TUNING REGISTER (ACCESS F9Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN(1)	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
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 **INTSRC:** Internal Oscillator Low-Frequency Source Select bit
1 = 31.25 kHz device clock derived from 8 MHz INTOSC source (divide-by-256 enabled)
0 = 31 kHz device clock derived directly from INTRC internal oscillator

bit 6 **PLLEN:** Frequency Multiplier Enable bit(1)
1 = 96 MHz PLL is enabled
0 = 96 MHz PLL is disabled

bit 5-0 **TUN<5:0>:** Frequency Tuning bits
011111 = Maximum frequency
011110
•
•
•
000001
000000 = Center frequency; oscillator module is running at the calibrated frequency
111111
•
•
•
100000 = Minimum frequency

Note 1: When the CFGPLLEN Configuration bit is used to enable the PLL, clearing OSCTUNE<6> will not disable the PLL.

3.3 Oscillator Settings for USB

When the PIC18F47J53 family devices are used for USB connectivity, a 6 MHz or 48 MHz clock must be provided to the USB module for operation in either Low-Speed or Full-Speed modes, respectively. This may require some forethought in selecting an oscillator frequency and programming the device.

The full range of possible oscillator configurations compatible with USB operation is shown in Table 3-5.

3.3.1 LOW-SPEED OPERATION

The USB clock for Low-Speed mode is derived from the primary oscillator or from the 96 MHz PLL. In order to operate the USB module in Low-Speed mode, a 6 MHz clock must be provided to the USB module.

See Table 3-4 and Table 3-5 for possible combinations which can be used for low-speed USB operation.

TABLE 3-4: CLOCK FOR LOW-SPEED USB

System Clock	CPDIV<1:0>	Microcontroller Clock	LS48MHZ	USB Clock
48	11	48 MHz	1	48/8 = 6 MHz
48	10	48/2 = 24 MHz	1	48/8 = 6 MHz
48	01	48/3 = 16 MHz	1	48/8 = 6 MHz
48	00	48/6 = 8 MHz	1	48/8 = 6 MHz
24	11	24 MHz	0	24/4 = 6 MHz
24	10	24/2 = 12 MHz	0	24/4 = 6 MHz
24	01	24/3 = 8 MHz	0	24/4 = 6 MHz
24	00	24/6 = 4 MHz	0	24/4 = 6 MHz

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
PMCONL	PIC18F2XJ53	PIC18F4XJ53	000- 0000	000- 0000	uuu- uuuu
PMMODEH	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMMODEL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDOUT2H	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDOUT2L	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDIN2H	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMDIN2L	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMEH	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMEL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
PMSTATH	PIC18F2XJ53	PIC18F4XJ53	00-- 0000	00-- 0000	uu-- uuuu
PMSTATL	PIC18F2XJ53	PIC18F4XJ53	10-- 1111	10-- 1111	uu-- uuuu
CVRCON	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
CCPTMRS0	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
CCPTMRS1	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
CCPTMRS2	PIC18F2XJ53	PIC18F4XJ53	0000 0000	0000 0000	uuuu uuuu
DSGPR1(6)	PIC18F2XJ53	PIC18F4XJ53	uuuu uuuu	uuuu uuuu	uuuu uuuu
DSGPR0(6)	PIC18F2XJ53	PIC18F4XJ53	uuuu uuuu	uuuu uuuu	uuuu uuuu
DSCONH(6)	PIC18F2XJ53	PIC18F4XJ53	0--- --00	0--- --00	u--- --uu
DSCONL(6)	PIC18F2XJ53	PIC18F4XJ53	---- -000	---- -000	---- -uuu
DSWAKEH(6)	PIC18F2XJ53	PIC18F4XJ53	---- ---q	---- ---0	---- ---u
DSWAKEL(6)	PIC18F2XJ53	PIC18F4XJ53	q-qq qq-q	0-00 00-0	u-uu uu-u
ANCON1	PIC18F2XJ53	PIC18F4XJ53	00-0 0000	uu-u uuuu	uu-u uuuu
ANCON0	PIC18F2XJ53	PIC18F4XJ53	0000 0000	uuuu uuuu	uuuu uuuu
ALRMCFG	PIC18F2XJ53	PIC18F4XJ53	0000 0000	uuuu uuuu	uuuu uuuu
ALRMRPT	PIC18F2XJ53	PIC18F4XJ53	0000 0000	uuuu uuuu	uuuu uuuu
ALRMVALH	PIC18F2XJ53	PIC18F4XJ53	xxxx xxxx	uuuu uuuu	uuuu uuuu
ALRMVALL	PIC18F2XJ53	PIC18F4XJ53	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON1	PIC18F2XJ53	PIC18F4XJ53	---- 0000	---- uuuu	---- uuuu
ODCON2	PIC18F2XJ53	PIC18F4XJ53	---- --00	---- --uu	---- --uu
ODCON3	PIC18F2XJ53	PIC18F4XJ53	---- --00	---- --uu	---- --uu
RTCCFG	PIC18F2XJ53	PIC18F4XJ53	0-00 0000	u-uu uuuu	u-uu uuuu
RTCCAL	PIC18F2XJ53	PIC18F4XJ53	0000 0000	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 5-1 for Reset value for specific condition.

5: Not implemented for PIC18F2XJ53 devices.

6: Not implemented for "LF" devices.

REGISTER 9-16: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3 (ACCESS FA5h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP
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 **SSP2IP:** Master Synchronous Serial Port 2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **BCL2IP:** Bus Collision Interrupt Priority bit (MSSP2 module)
1 = High priority
0 = Low priority

bit 5 **RC2IP:** EUSART2 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TX2IP:** EUSART2 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **TMR4IE:** TMR4 to PR4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CTMUIP:** Charge Time Measurement Unit (CTMU) Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3GIP:** Timer3 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RTCCIP:** RTCC Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 10-8: RPINR3: PERIPHERAL PIN SELECT INPUT REGISTER 3 (BANKED EE3h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	INTR3R4	INTR3R3	INTR3R2	INTR3R1	INTR3R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **INTR3R<4:0>:** Assign External Interrupt 3 (INT3) to the Corresponding RPn Pin bits

REGISTER 10-9: RPINR4: PERIPHERAL PIN SELECT INPUT REGISTER 4 (BANKED EE4h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	T0CKR4	T0CKR3	T0CKR2	T0CKR1	T0CKR0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **T0CKR<4:0>:** Timer0 External Clock Input (T0CKI) to the Corresponding RPn Pin bits

REGISTER 10-10: RPINR6: PERIPHERAL PIN SELECT INPUT REGISTER 6 (BANKED EE6h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **T3CKR<4:0>:** Timer3 External Clock Input (T3CKI) to the Corresponding RPn Pin bits

REGISTER 10-17: RPINR16: PERIPHERAL PIN SELECT INPUT REGISTER 16 (BANKED EF7h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	RX2DT2R4	RX2DT2R3	RX2DT2R2	RX2DT2R1	RX2DT2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RX2DT2R<4:0>:** EUSART2 Synchronous/Asynchronous Receive (RX2/DT2) to the Corresponding RPn Pin bits

REGISTER 10-18: RPINR15: PERIPHERAL PIN SELECT INPUT REGISTER 15 (BANKED EE7h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **T5CKR<4:0>:** Timer5 External Clock Input (T5CKI) to the Corresponding RPn Pin bits

REGISTER 10-19: RPINR17: PERIPHERAL PIN SELECT INPUT REGISTER 17 (BANKED EF8h)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	CK2R4	CK2R3	CK2R2	CK2R1	CK2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **CK2R<4:0>:** EUSART2 Clock Input (CK2) to the Corresponding RPn Pin bits

REGISTER 10-20: RPINR21: PERIPHERAL PIN SELECT INPUT REGISTER 21 (BANKED EFCh)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **SDI2R<4:0>:** Assign SPI2 Data Input (SDI2) to the Corresponding RPn Pin bits

REGISTER 10-24: RPOR0: PERIPHERAL PIN SELECT OUTPUT REGISTER 0 (BANKED EC1h)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RP0R<4:0>:** Peripheral Output Function is Assigned to RP0 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-25: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTER 1 (BANKED EC7h)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RP1R<4:0>:** Peripheral Output Function is Assigned to RP1 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-26: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2 (BANKED EC3h)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RP2R<4:0>:** Peripheral Output Function is Assigned to RP2 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-39: RPOR18: PERIPHERAL PIN SELECT OUTPUT REGISTER 18 (BANKED ED2h)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RP18R<4:0>:** Peripheral Output Function is Assigned to RP18 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-40: RPOR19: PERIPHERAL PIN SELECT OUTPUT REGISTER 19 (BANKED ED3h)(1)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RP19R<4:0>:** Peripheral Output Function is Assigned to RP19 Output Pin bits
(see Table 10-14 for peripheral function numbers)

Note 1: RP19 pins are not available on 28-pin devices.

REGISTER 10-41: RPOR20: PERIPHERAL PIN SELECT OUTPUT REGISTER 20 (BANKED ED4h)(1)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit if IOLOCK = 0		
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 4-0 **RP20R<4:0>:** Peripheral Output Function is Assigned to RP20 Output Pin bits
(see Table 10-14 for peripheral function numbers)

Note 1: RP20 pins are not available on 28-pin devices.

REGISTER 11-9: PMADDRH: PARALLEL PORT ADDRESS REGISTER HIGH BYTE (MASTER MODES ONLY) (ACCESS F6Fhh)(1)

U0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CS1	Parallel Master Port Address High Byte<13:8>					
bit 7							bit 0

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

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

bit 6 **CS1:** Chip Select bit
If PMCON<7:6> = 10:
1 = Chip select is active
0 = Chip select is inactive
If PMCON<7:6> = 11 or 00:
Bit functions as ADDR<14>.

bit 5-0 **Parallel Master Port Address:** High Byte<13:8> bits

Note 1: In Enhanced Slave mode, PMADDRH functions as PMDOUT1H, one of the Output Data Buffer registers.

REGISTER 11-10: PMADDRL: PARALLEL PORT ADDRESS REGISTER LOW BYTE (MASTER MODES ONLY) (ACCESS F6Eh)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parallel Master Port Address Low Byte<7:0>							
bit 7							bit 0

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

bit 7-0 **Parallel Master Port Address:** Low Byte<7:0> bits

Note 1: In Enhanced Slave mode, PMADDRL functions as PMDOUT1L, one of the Output Data Buffer registers.

TABLE 11-2: REGISTERS ASSOCIATED WITH PMP MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PMPIF(2)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIE1	PMPIE(2)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
IPR1	PMPIP(2)	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
PMCONH(2)	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN
PMCONL(2)	CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP
PMADDRH(1,2)/	—	CS1	Parallel Master Port Address High Byte					
PMDOUT1H(1,2)	Parallel Port Out Data High Byte (Buffer 1)							
PMADDRL(1,2)/	Parallel Master Port Address Low Byte							
PMDOUT1L(1,2)	Parallel Port Out Data Low Byte (Buffer 0)							
PMDOUT2H(2)	Parallel Port Out Data High Byte (Buffer 3)							
PMDOUT2L(2)	Parallel Port Out Data Low Byte (Buffer 2)							
PMDIN1H(2)	Parallel Port In Data High Byte (Buffer 1)							
PMDIN1L(2)	Parallel Port In Data Low Byte (Buffer 0)							
PMDIN2H(2)	Parallel Port In Data High Byte (Buffer 3)							
PMDIN2L(2)	Parallel Port In Data Low Byte (Buffer 2)							
PMMODEH(2)	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
PMMODEL(2)	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0
PMEH(2)	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
PMEL(2)	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
PMSTATH(2)	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
PMSTATL(2)	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
PADCFG1	—	—	—	—	—	RTSECSEL1	RTSECSEL0	PMPTTL

Legend: — = unimplemented, read as '0'. Shaded cells are not used during PMP operation.

Note 1: The PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode.

2: These bits and/or registers are only available in 44-pin devices.

12.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable. Its value is set by the PSA and T0PS<2:0> bits (T0CON<3:0>), which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When it is assigned, prescale values from 1:2 through 1:256, in power-of-2 increments, are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) clear the prescaler count.

Note:	Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

12.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed “on-the-fly” during program execution.

12.4 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit (INTCON<5>). Before re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine (ISR).

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 12-1: REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR0L	Timer0 Register Low Byte							
TMR0H	Timer0 Register High Byte							
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

Legend: Shaded cells are not used by Timer0.

REGISTER 18-3: CCPTMRS2: CCP4-10 TIMER SELECT 2 REGISTER (BANKED F50h)

U-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	C10TSEL0	—	C9TSEL0	C8TSEL1	C8TSEL0
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 **C10TSEL0:** CCP10 Timer Selection bit
0 = CCP10 is based off of TMR1/TMR2
1 = Reserved; do not use

bit 3 **Unimplemented:** Read as '0'

bit 2 **C9TSEL0:** CCP9 Timer Selection bit
0 = CCP9 is based off of TMR1/TMR2
1 = CCP9 is based off of TMR1/TMR4

bit 1-0 **C8TSEL<1:0>:** CCP8 Timer Selection bits
00 = CCP8 is based off of TMR1/TMR2
01 = CCP8 is based off of TMR1/TMR4
10 = CCP8 is based off of TMR1/TMR6
11 = Reserved; do not use

REGISTER 19-1: CCPxCON: ECCP1/2/3 CONTROL (1, ACCESS FBAh; 2, FB4h; 3, BANKED F15h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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 7-6 **PxM<1:0>:** Enhanced PWM Output Configuration bits

If CCPxM<3:2> = 00, 01, 10:

xx = PxA assigned as capture/compare input/output; PxB, PxC and PxD assigned as port pins

If CCPxM<3:2> = 11:

00 = Single output: PxA, PxB, PxC and PxD are controlled by steering (see **Section 19.4.7 "Pulse Steering Mode"**)

01 = Full-bridge output forward: PxD modulated; PxA active; PxB, PxC inactive

10 = Half-bridge output: PxA, PxB modulated with dead-band control; PxC and PxD assigned as port pins

11 = Full-bridge output reverse: PxB modulated; PxC active; PxA and PxD inactive

bit 5-4 **DCxB<1:0>**: PWM Duty Cycle bit 1 and bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in ECCPRxL.

bit 3-0 **CCPxM<3:0>**: ECCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCPx module)

0001 = Reserved

0010 = Compare mode, toggle output on match

0011 = Capture mode

0100 = Capture mode; every falling edge

0101 = Capture mode; every rising edge

0110 = Capture mode; every fourth rising edge

0111 = Capture mode; every 16th rising edge

1000 = Compare mode; initialize ECCPx pin low, set output on compare match (set CCPxIF)

1001 = Compare mode; initialize ECCPx pin high, clear output on compare match (set CCPxIF)

1010 = Compare mode; generate software interrupt only, ECCPx pin reverts to I/O state

1011 = Compare mode; trigger special event (ECCPx resets TMR1 or TMR3, starts A/D conversion, sets CCPxIF bit)

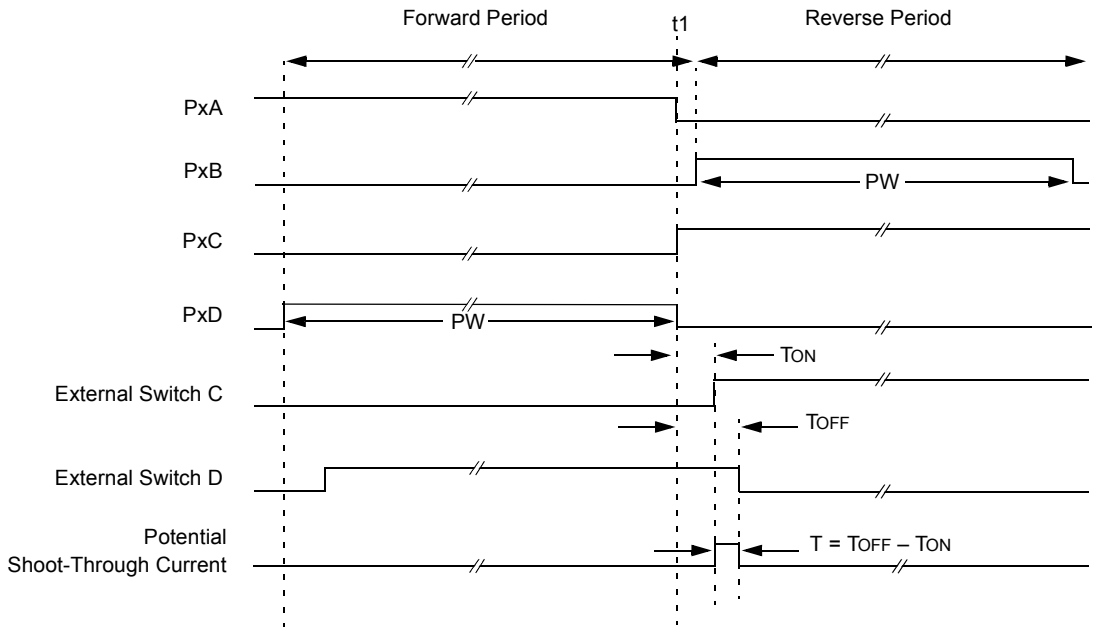
1100 = PWM mode; PxA and PxC active-high; PxB and PxD active-high

1101 = PWM mode; PxA and PxC active-high; PxB and PxD active-low

1110 = PWM mode; PxA and PxC active-low; PxB and PxD active-high

1111 = PWM mode; PxA and PxC active-low; PxB and PxD active-low

FIGURE 19-11: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



Note 1: All signals are shown as active-high.

2: TON is the turn-on delay of power switch, QC, and its driver.

3: TOFF is the turn-off delay of power switch, QD, and its driver.

19.4.3 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

Note:	When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

The CCPxM<1:0> bits of the CCPxCON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (PxA/PxC and PxB/PxD). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity configuration while the PWM pin output drivers are enabled is not recommended, since it may result in damage to the application circuits.

The PxA, PxB, PxC and PxD output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMR2IF or TMR4IF bit of the PIR1 or PIR3 register being set as the second PWM period begins.

19.4.4 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the ECCPxAS<2:0> bits (ECCPxAS<6:4>). A shutdown event may be generated by:

- A logic '0' on the pin that is assigned to the FLT0 input function
- Comparator C1
- Comparator C2
- Setting the ECCPxASE bit in firmware

A shutdown condition is indicated by the ECCPxASE (Auto-Shutdown Event Status) bit (ECCPxAS<7>). If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

20.3.1 REGISTERS

Each MSSP module has four registers for SPI mode operation. These are:

- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Status Register (SSPxSTAT)
- Serial Receive/Transmit Buffer Register (SSPxBUF)
- MSSPx Shift Register (SSPxSR) – Not directly accessible

SSPxCON1 and SSPxSTAT are the control and status registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

SSPxSR is the shift register used for shifting data in or out. SSPxBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPxSR and SSPxBUF together, create a double-buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not double-buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

REGISTER 20-1: SSPxSTAT: MSSPx STATUS REGISTER (SPI MODE) (ACCESS 1, FC7h; 2, F73h)

R/W-1	R/W-1	R-1	R-1	R-1	R-1	R-1	R-1
SMP	CKE(1)	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
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 **SMP:** Sample bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 6 **CKE:** SPI Clock Select bit(1)
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit
Used in I^2C mode only.

bit 4 **P:** Stop bit
Used in I^2C mode only; this bit is cleared when the MSSP module is disabled, SSPEN is cleared.

bit 3 **S:** Start bit
Used in I^2C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit
Used in I^2C mode only.

bit 1 **UA:** Update Address bit
Used in I^2C mode only.

bit 0 **BF:** Buffer Full Status bit
1 = Receive complete, SSPxBUF is full
0 = Receive not complete, SSPxBUF is empty

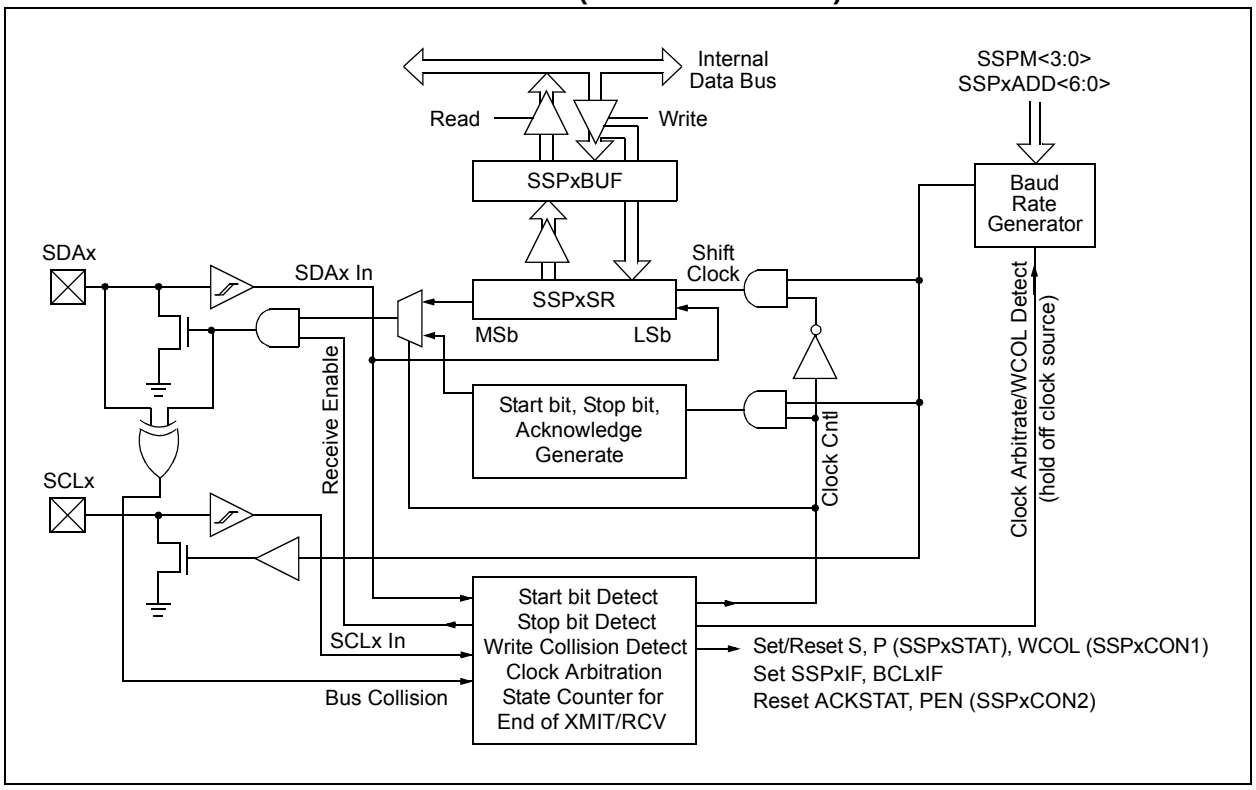
Note 1: Polarity of the clock state is set by the CKP bit (SSPxCON1<4>).

Note:	The MSSP module, when configured in I^2C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.

The following events will cause the MSSP Interrupt Flag bit, SSPxIF, to be set (and MSSP interrupt, if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmitted
- Repeated Start

FIGURE 20-18: MSSPx BLOCK DIAGRAM (I^2C MASTER MODE)



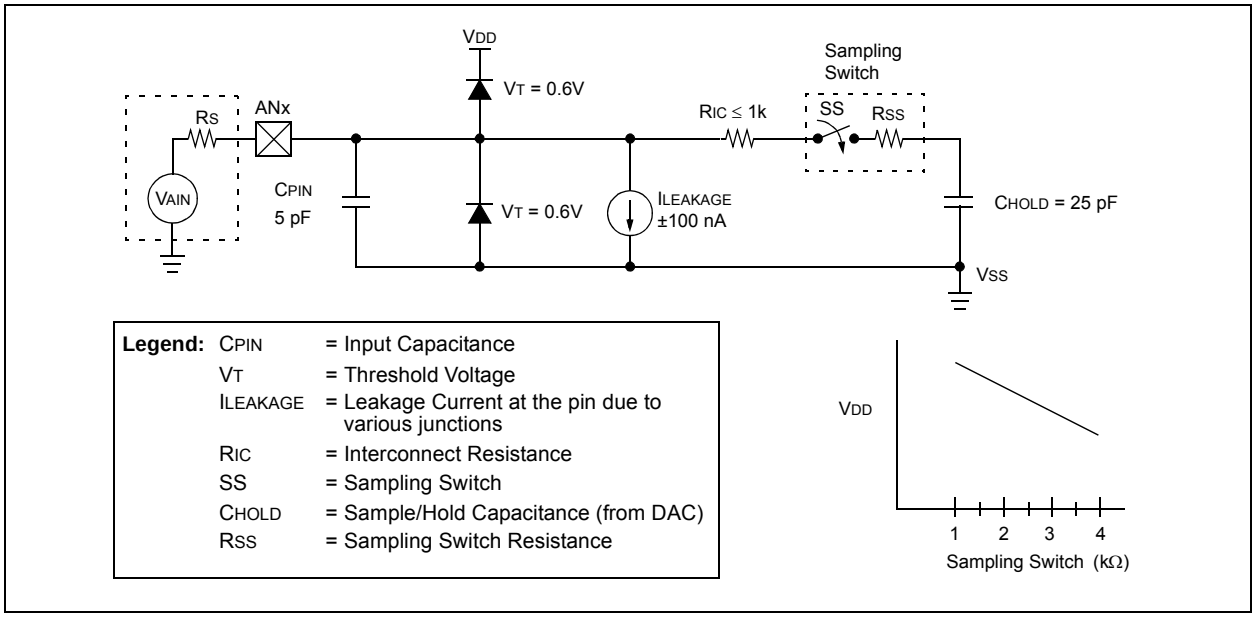
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs. To determine acquisition time, see **Section 22.1 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/$\overline{\text{DONE}}$ bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

1. Configure the A/D module:
   - Configure the required ADC pins as analog pins using ANCON0, ANCON1
   - Set voltage reference using ADCON0
   - Select A/D input channel (ADCON0)
   - Select A/D acquisition time (ADCON1)
   - Select A/D conversion clock (ADCON1)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time (if required).
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear bit, ADIF, if required.
7. For the next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum Wait of 2 TAD is required before the next acquisition starts.

FIGURE 22-2: ANALOG INPUT MODEL



23.0 UNIVERSAL SERIAL BUS (USB)

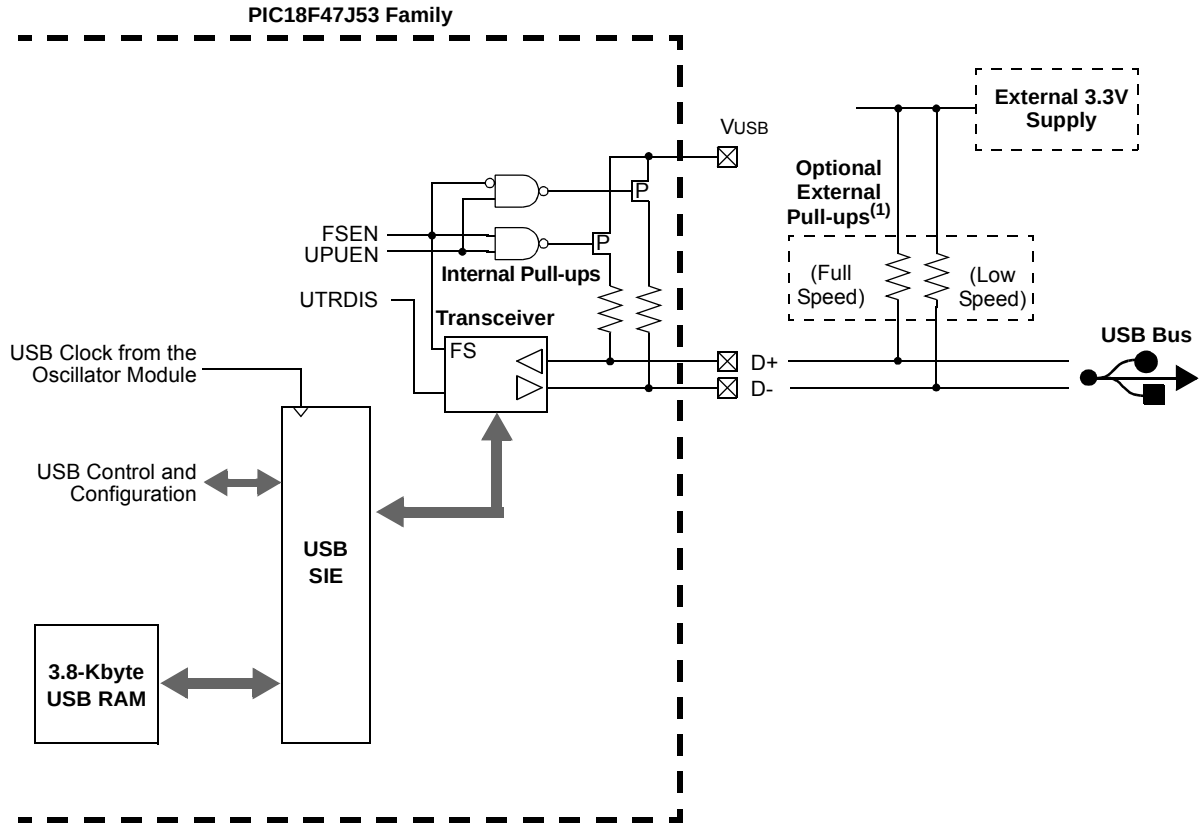
This section describes the details of the USB peripheral. Because of the very specific nature of the module, knowledge of USB is expected. Some high-level USB information is provided in **Section 23.9 "Overview of USB"** only for application design reference. Designers are encouraged to refer to the official specification published by the USB Implementers Forum (USB-IF) for the latest information. USB Specification Revision 2.0 is the most current specification at the time of publication of this document.

23.1 Overview of the USB Peripheral

PIC18F47J53 family devices contain a full-speed and low-speed, compatible USB Serial Interface Engine (SIE) that allows fast communication between any USB host and the PIC® MCU. The SIE can be interfaced directly to the USB, utilizing the internal transceiver.

Some special hardware features have been included to improve performance. Dual access port memory in the device's data memory space (USB RAM) has been supplied to share direct memory access between the microcontroller core and the SIE. Buffer descriptors are also provided, allowing users to freely program end-point memory usage within the USB RAM space. Figure 23-1 provides a general overview of the USB peripheral and its features.

FIGURE 23-1: USB PERIPHERAL AND OPTIONS



Note 1: The internal pull-up resistors should be disabled (UPUEN = 0) if external pull-up resistors are used.

23.6.4 USB TRANSCEIVER CURRENT CONSUMPTION

The USB transceiver consumes a variable amount of current depending on the characteristic impedance of the USB cable, the length of the cable, the VUSB supply voltage and the actual data patterns moving across the USB cable. Longer cables have larger capacitances and consume more total energy when switching output states.

Data patterns that consist of “IN” traffic consume far more current than “OUT” traffic. IN traffic requires the PIC® MCU to drive the USB cable, whereas OUT traffic requires that the host drive the USB cable.

The data that is sent across the USB cable is NRZI encoded. In the NRZI encoding scheme, ‘0’ bits cause a toggling of the output state of the transceiver (either from a “J” state to a “K” state or vise versa). With the exception of the effects of bit stuffing, NRZI encoded ‘1’ bits do not cause the output state of the transceiver to change. Therefore, IN traffic consisting of data bits of value, ‘0’, cause the most current consumption, as the transceiver must charge/discharge the USB cable in order to change states.

More details about NRZI encoding and bit stuffing can be found in the USB 2.0 Specification’s Section 7.1, although knowledge of such details is not required to make USB applications using the PIC18F47J53 family of microcontrollers. Among other things, the SIE handles bit stuffing/unstuffing, NRZI encoding/decoding and CRC generation/checking in hardware.

The total transceiver current consumption will be application-specific. However, to help estimate how much current actually may be required in full-speed applications, Equation 23-1 can be used.

See Equation 23-2 to know how this equation can be used for a theoretical application.

EQUATION 23-1: ESTIMATING USB TRANSCEIVER CURRENT CONSUMPTION

$$I_{XCVR} = \frac{(40\ mA \bullet V_{USB} \bullet P_{ZERO} \bullet P_{IN} \bullet L_{CABLE})}{(3.3V \bullet 5m)} + I_{PULLUP}$$

Legend: VUSB – Voltage applied to the VUSB pin in volts (should be 3.0V to 3.6V).

PZERO – Percentage (in decimal) of the IN traffic bits sent by the PIC® MCU that are a value of ‘0’.

PIN – Percentage (in decimal) of total bus bandwidth that is used for IN traffic.

LCABLE – Length (in meters) of the USB cable. The USB 2.0 Specification requires that full-speed applications use cables no longer than 5m.

IPULLUP – Current which the nominal, 1.5 kΩ pull-up resistor (when enabled) must supply to the USB cable. On the host or hub end of the USB cable, 15 kΩ nominal resistors (14.25 kΩ to 24.8 kΩ) are present which pull both the D+ and D- lines to ground. During bus Idle conditions (such as between packets or during USB Suspend mode), this results in up to 218 μA of quiescent current drawn at 3.3V.

IPULLUP is also dependant on bus traffic conditions and can be as high as 2.2 mA when the USB bandwidth is fully utilized (either IN or OUT traffic) for data that drives the lines to the “K” state most of the time.

FIGURE 31-14: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)

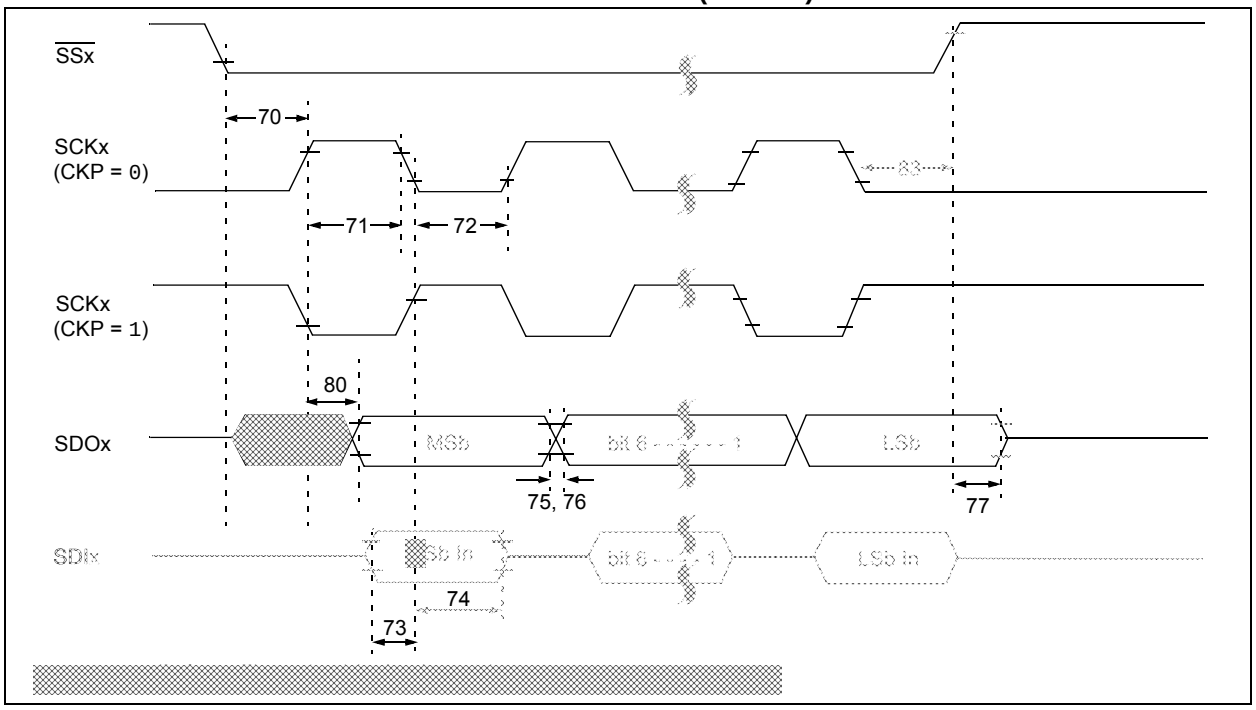


TABLE 31-22: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 Tcy	—	ns	
70A	TssL2WB	$\overline{SSx}$ ↓ to Write to SSPxBUF		3 Tcy	—	ns	
71	TscH	SCKx Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
71A			Single byte	40	—	ns	**(Note 1)**
72	TscL	SCKx Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
72A			Single byte	40	—	ns	**(Note 1)**
73	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge		25	—	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge		35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
				100	—	ns	VDD = 2.15V
75	TdoR	SDOx Data Output Rise Time		—	25	ns	PORTB or PORTC
76	TdoF	SDOx Data Output Fall Time		—	25	ns	PORTB or PORTC
77	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance		10	70	ns	
80	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge		—	50	ns	VDD = 3.3V, VDDCORE = 2.5V
				—	100	ns	VDD = 2.15V
83	TscH2ssH, TscL2ssH	$\overline{SSx}$ ↑ after SCKx Edge		1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter #73A.
2: Only if Parameter #71A and #72A are used.