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
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f26j13t-i-ss

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
IPR1	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu(3)
PIE1	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
RCSTA2	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
OSCTUNE	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
T1GCON	PIC18F2XJ13	PIC18F4XJ13	0000 0x00	0000 0x00	uuuu uxuu
T3GCON	PIC18F2XJ13	PIC18F4XJ13	0000 0x00	uuuu uxuu	uuuu uxuu
TRISE(5)	—	PIC18F4XJ13	00-- -111	00-- -111	uu-- -uuu
TRISD(5)	—	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
TRISA	PIC18F2XJ13	PIC18F4XJ13	111- 1111	111- 1111	uuu- uuuu
PIE5	PIC18F2XJ13	PIC18F4XJ13	--00 0000	--00 0000	--uu uuuu
IPR4	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
PIR4	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PIE4	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
LATE(5)	—	PIC18F4XJ13	---- -xxx	---- -uuu	---- -uuu
LATD(5)	—	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA	PIC18F2XJ13	PIC18F4XJ13	xxx- xxxx	uuu- uuuu	uuu- uuuu
DMACON1	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
OSCCON2	PIC18F2XJ13	PIC18F4XJ13	-0-1 01--	-0-1 u1--	-u-u uu--
DMACON2	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
HLVDCON	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PORTE(5)	—	PIC18F4XJ13	---- -xxx	---- -uuu	---- -uuu
PORTD(5)	—	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	PIC18F2XJ13	PIC18F4XJ13	xxx- xxxx	uuu- uuuu	uuu- uuuu
SPBRGH1	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	PIC18F2XJ13	PIC18F4XJ13	0100 0-00	0100 0-00	uuuu u-uu

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 the Reset value for a specific condition.

5: Not implemented on PIC18F2XJ13 devices.

6: Not implemented on "LF" devices.

6.3.4 SPECIAL FUNCTION REGISTERS

The SFRs are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. SFRs start at the top of data memory (FFFh) and extend downward to occupy more than the top half of Bank 15 (F40h to FFFh). Table 6-2, Table 6-3 and Table 6-4 provide a list of these registers.

The SFRs can be classified into two sets: those associated with the "core" device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and Interrupt registers are described in their corresponding chapters, while the ALU's STATUS register is described later in this section. Registers related to the operation of the peripheral features are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as '0's

Note: The SFRs located between EB0h and F5Fh are not part of the Access Bank. Either BANKED instructions (using BSR) or the MOVFF instruction should be used to access these locations. When programming in MPLAB® C18, the compiler will automatically use the appropriate addressing mode.

TABLE 6-2: ACCESS BANK SPECIAL FUNCTION REGISTER MAP

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(1)	FBFh	PSTR1CON	F9Fh	IPR1	F7Fh	SPBRGH1
FFEh	TOSH	FDEh	POSTINC2(1)	FBEh	ECCP1AS	F9Eh	PIR1	F7Eh	BAUDCON1
FFDh	TOSL	FDDh	POSTDEC2(1)	FBDh	ECCP1DEL	F9Dh	PIE1	F7Dh	SPBRGH2
FFCh	STKPTR	FDCh	PREINC2(1)	FBCh	CCPR1H	F9Ch	RCSTA2	F7Ch	BAUDCON2
FFBh	PCLATU	FDBh	PLUSW2(1)	FBBh	CCPR1L	F9Bh	OSCTUNE	F7Bh	TMR3H
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP1CON	F9Ah	T1GCON	F7Ah	TMR3L
FF9h	PCL	FD9h	FSR2L	FB9h	PSTR2CON	F99h	IPR5	F79h	T3CON
FF8h	TBLPTRU	FD8h	STATUS	FB8h	ECCP2AS	F98h	PIR5	F78h	TMR4
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP2DEL	F97h	T3GCON	F77h	PR4
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	CCPR2H	F96h	TRISE	F76h	T4CON
FF5h	TABLAT	FD5h	T0CON	FB5h	CCPR2L	F95h	TRISD	F75h	SSP2BUF
FF4h	PRODH	FD4h	—(5)	FB4h	CCP2CON	F94h	TRISC	F74h	SSP2ADD(3)
FF3h	PRODL	FD3h	OSCCON	FB3h	CTMUCONH	F93h	TRISB	F73h	SSP2STAT
FF2h	INTCON	FD2h	CM1CON	FB2h	CTMUCONL	F92h	TRISA	F72h	SSP2CON1
FF1h	INTCON2	FD1h	CM2CON	FB1h	CTMUICON	F91h	PIE5	F71h	SSP2CON2
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	IPR4	F70h	CMSTAT
FEFh	INDF0(1)	FCFh	TMR1H	FAFh	RCREG1	F8Fh	PIR4	F6Fh	PMADDRH(2,4)
FEEh	POSTINC0(1)	FCEh	TMR1L	FAEh	TXREG1	F8Eh	PIE4	F6Eh	PMADDRL(2,4)
FEDh	POSTDEC0(1)	FCDh	T1CON	FADh	TXSTA1	F8Dh	LATE(2)	F6Dh	PMDIN1H(2)
FECh	PREINC0(1)	FCCh	TMR2	FACh	RCSTA1	F8Ch	LATD(2)	F6Ch	PMDIN1L(2)
FEBh	PLUSW0(1)	FCBh	PR2	FABh	SPBRG2	F8Bh	LATC	F6Bh	TXADDRL
FEAh	FSR0H	FCAh	T2CON	FAAh	RCREG2	F8Ah	LATB	F6Ah	TXADDRH
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	TXREG2	F89h	LATA	F69h	RXADDRL
FE8h	WREG	FC8h	SSP1ADD(3)	FA8h	TXSTA2	F88h	DMACON1	F68h	RXADDRH
FE7h	INDF1(1)	FC7h	SSP1STAT	FA7h	EECON2	F87h	OSCCON2(5)	F67h	DMABCL
FE6h	POSTINC1(1)	FC6h	SSP1CON1	FA6h	EECON1	F86h	DMACON2	F66h	DMABCH
FE5h	POSTDEC1(1)	FC5h	SSP1CON2	FA5h	IPR3	F85h	HLVDCON	F65h	—
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE(2)	F64h	—
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD(2)	F63h	—
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	—
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	—
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	—

Note 1: This is not a physical register.
2: This register is not available on 28-pin devices.
3: SSPxADD and SSPxMSK share the same address.
4: PMADDRH and PMDOUTH share the same address, and PMADDRL and PMDOUTL share the same address. PMADDRx is used in Master modes and PMDOUTx is used in Slave modes.
5: Reserved; do not write to this location.

6.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- POSTDEC: accesses the FSR value, then automatically decrements it by one thereafter
- POSTINC: accesses the FSR value, then automatically increments it by one thereafter
- PREINC: increments the FSR value by one, then uses it in the operation
- PLUSW: adds the signed value of the W register (range of -128 to +127) to that of the FSR and uses the new value in the operation

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by the value in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, roll-overs of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

6.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations. As a specific case, assume that FSR0H:FSR0L contains FE7h, the address of INDF1. Attempts to read the value of the INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a `NOP`.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise appropriate caution that they do not inadvertently change settings that might affect the operation of the device.

6.5 Program Memory and the Extended Instruction Set

The operation of program memory is unaffected by the use of the extended instruction set.

Enabling the extended instruction set adds five additional two-word commands to the existing PIC18 instruction set: `ADDFSR`, `CALLW`, `MOVSF`, `MOVSS` and `SUBFSR`. These instructions are executed as described in **Section 6.2.4 “Two-Word Instructions”**.

7.5 Writing to Flash Program Memory

The programming block is 32 words or 64 bytes. Programming one word or 2 bytes at a time is also supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 64 holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation (if WPROG = 0). All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

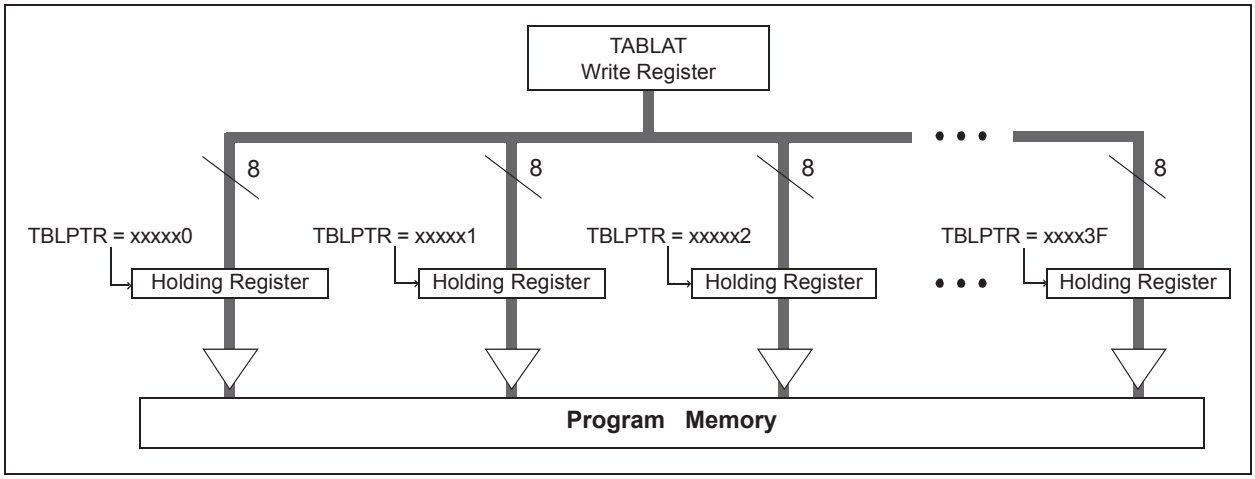
The long write is necessary for programming the internal Flash. Instruction execution is Halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

Note 1: Unlike previous PIC® devices, devices of the PIC18F47J13 Family do not reset the holding registers after a write occurs. The holding registers must be cleared or overwritten before a programming sequence.

2: To maintain the endurance of the program memory cells, each Flash byte should not be programmed more than once between erase operations. Before attempting to modify the contents of the target cell a second time, an erase of the target page, or a bulk erase of the entire memory, must be performed.

FIGURE 7-5: TABLE WRITES TO FLASH PROGRAM MEMORY



7.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 1024 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load the Table Pointer register with address being erased.
4. Execute the erase procedure.
5. Load the Table Pointer register with the address of the first byte being written, minus 1.
6. Write the 64 bytes into the holding registers with auto-increment.
7. Set the WREN bit (EECON1<2>) to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit; this will begin the write cycle.
12. The CPU will stall for the duration of the write for TIW (see parameter D133A).
13. Re-enable interrupts.
14. Repeat Steps 6 through 13 until all 1024 bytes are written to program memory.
15. Verify the memory (table read).

An example of the required code is provided in Example 7-3 on the following page.

Note: Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 64 bytes in the holding register.

REGISTER 9-7: PIR4: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 4 (ACCESS F8Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCP10IF	CCP9IF	CCP8IF	CCP7IF	CCP6IF	CCP5IF	CCP4IF	CCP3IF
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 7-1 **CCP10IF:CCP4IF:** CCP<10:4> Interrupt Flag bits

Capture Mode
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare Mode
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM Mode
Unused in this mode.

bit 0 **CCP3IF:** ECCP3 Interrupt Flag bit

Capture Mode
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare Mode
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM Mode
Unused in this mode.

REGISTER 10-30: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6 (BANKED EC6h)

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
—	—	—	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
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 **RP6R<4:0>:** Peripheral Output Function is Assigned to RP6 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-31: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7 (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
—	—	—	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
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 **RP7R<4:0>:** Peripheral Output Function is Assigned to RP7 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-32: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8 (BANKED EC8h)

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
—	—	—	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
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 **RP8R<4:0>:** Peripheral Output Function is Assigned to RP8 Output Pin bits
(see Table 10-14 for peripheral function numbers)

NOTES:

TABLE 17-3: RTCVALH AND RTCVALL REGISTER MAPPING

RTCPTR<1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALRMCFG<1:0>) to select the desired Alarm register pair.

By reading or writing to the ALRMVALH register, the Alarm Pointer value, ALRMPTR<1:0>, decrements by 1 until it reaches '00'. Once it reaches '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 17-4: ALRMVAL REGISTER MAPPING

ALRMPTR<1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

17.2.9 CALIBRATION

The real-time crystal input can be calibrated using the periodic auto-adjust feature. When properly calibrated, the RTCC can provide an error of less than three seconds per month.

To perform this calibration, find the number of error clock pulses and store the value in the lower half of the RTCCAL register. The 8-bit, signed value, loaded into RTCCAL, is multiplied by four and will either be added or subtracted from the RTCC timer, once every minute.

To calibrate the RTCC module:

1. Use another timer resource on the device to find the error of the 32.768 kHz crystal.
2. Convert the number of error clock pulses per minute (see Equation 17-1).

EQUATION 17-1: CONVERTING ERROR CLOCK PULSES

(Ideal Frequency (32,768) – Measured Frequency) * 60 = Error Clocks per Minute

   - If the oscillator is *faster* than ideal (negative result from Step 2), the RTCCFG register value needs to be negative. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.
   - If the oscillator is *slower* than ideal (positive result from Step 2), the RTCCFG register value needs to be positive. This causes the specified number of clock pulses to be added to the timer counter, once every minute.
3. Load the RTCCAL register with the correct value.

Writes to the RTCCAL register should occur only when the timer is turned off, or immediately after the rising edge of the seconds pulse.

Note: In determining the crystal's error value, it is the user's responsibility to include the crystal's initial error from drift due to temperature or crystal aging.

REGISTER 19-6: PSTRxCON: PULSE STEERING CONTROL (1, ACCESS FBFh; 2, FB9h; 3, BANKED F1Ah)(1)

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA
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 **CMPL<1:0>:** Complementary Mode Output Assignment Steering Sync bits
1 = Modulated output pin toggles between PxA and PxB for each period
0 = Complementary output assignment is disabled; the STRD:STRA bits are used to determine Steering mode

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

bit 4 **STRSYNC:** Steering Sync bit
1 = Output steering update occurs on the next PWM period
0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STRD:** Steering Enable D bit
1 = PxD pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxD pin is assigned to a port pin

bit 2 **STRC:** Steering Enable C bit
1 = PxC pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxC pin is assigned to a port pin

bit 1 **STRB:** Steering Enable B bit
1 = PxB pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxB pin is assigned to a port pin

bit 0 **STRA:** Steering Enable A bit
1 = PxA pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxA pin is assigned to a port pin

Note 1: The PWM Steering mode is available only when the CCPxCON register bits, CCPxM<3:2> = 11 and PxM<1:0> = 00.

20.3.9 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in Full-Power mode. In the case of Sleep mode, all clocks are Halted.

In Idle modes, a clock is provided to the peripherals. That clock can be from the primary clock source, the secondary clock (Timer1 oscillator) or the INTOSC source. See **Section 3.3 "Clock Sources and Oscillator Switching"** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are Halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSPx interrupt flag bit will be set, and if enabled, will wake the device.

20.3.10 EFFECTS OF A RESET

A Reset disables the MSSP modules and terminates the current transfer.

20.3.11 BUS MODE COMPATIBILITY

Table 20-1 provides the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 20-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit, which controls when the data is sampled.

20.3.12 SPI CLOCK SPEED AND MODULE INTERACTIONS

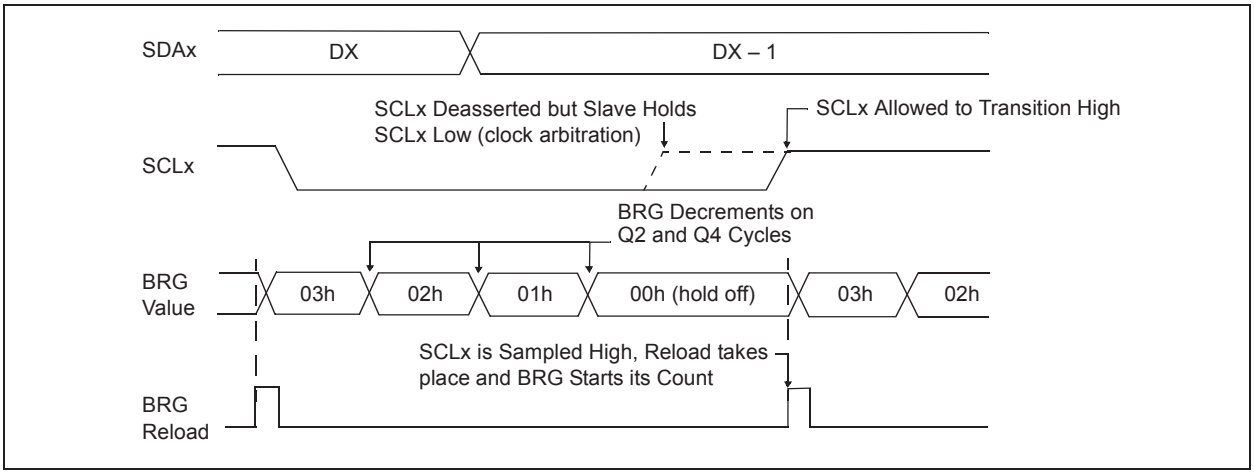
Because MSSP1 and MSSP2 are independent modules, they can operate simultaneously at different data rates. Setting the SSPM<3:0> bits of the SSPx-CON1 register determines the rate for the corresponding module.

An exception is when both modules use Timer2 as a time base in Master mode. In this instance, any changes to the Timer2 module's operation will affect both MSSP modules equally. If different bit rates are required for each module, the user should select one of the other three time base options for one of the modules.

20.5.7.2 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCLx pin (SCLx allowed to float high). When the SCLx pin is allowed to float high, the BRG is suspended from counting until the SCLx pin is actually sampled high. When the SCLx pin is sampled high, the BRG is reloaded with the contents of SSPxADD<6:0> and begins counting. This ensures that the SCLx high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 20-20).

FIGURE 20-20: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



20.5.8 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN (SSPxCON2<0>). If the SDAx and SCLx pins are sampled high, the BRG is reloaded with the contents of SSPxADD<6:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the Start bit (SSPxSTAT<3>) to be set. Following this, the BRG is reloaded with the contents of SSPxADD<6:0> and resumes its count. When the BRG times out (TBRG), the SEN bit (SSPxCON2<0>) will be automatically cleared by hardware. The BRG is suspended, leaving the SDAx line held low and the Start condition is complete.

Note: If, at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

20.5.8.1 WCOL Status Flag

If the user writes the SSPxBUF when a Start sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

Note: Because queueing of events is not allowed, writing to the lower five bits of SSPxCON2 is disabled until the Start condition is complete.

FIGURE 20-21: FIRST START BIT TIMING

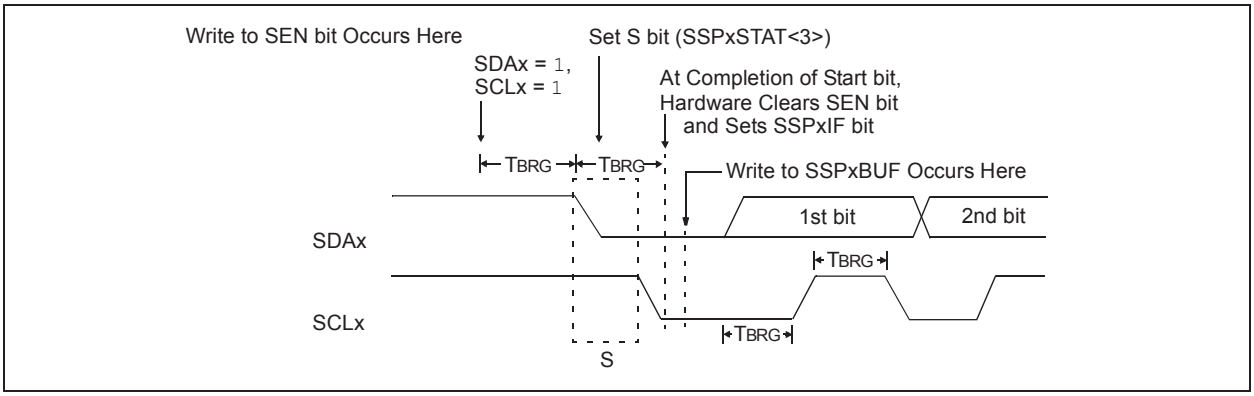
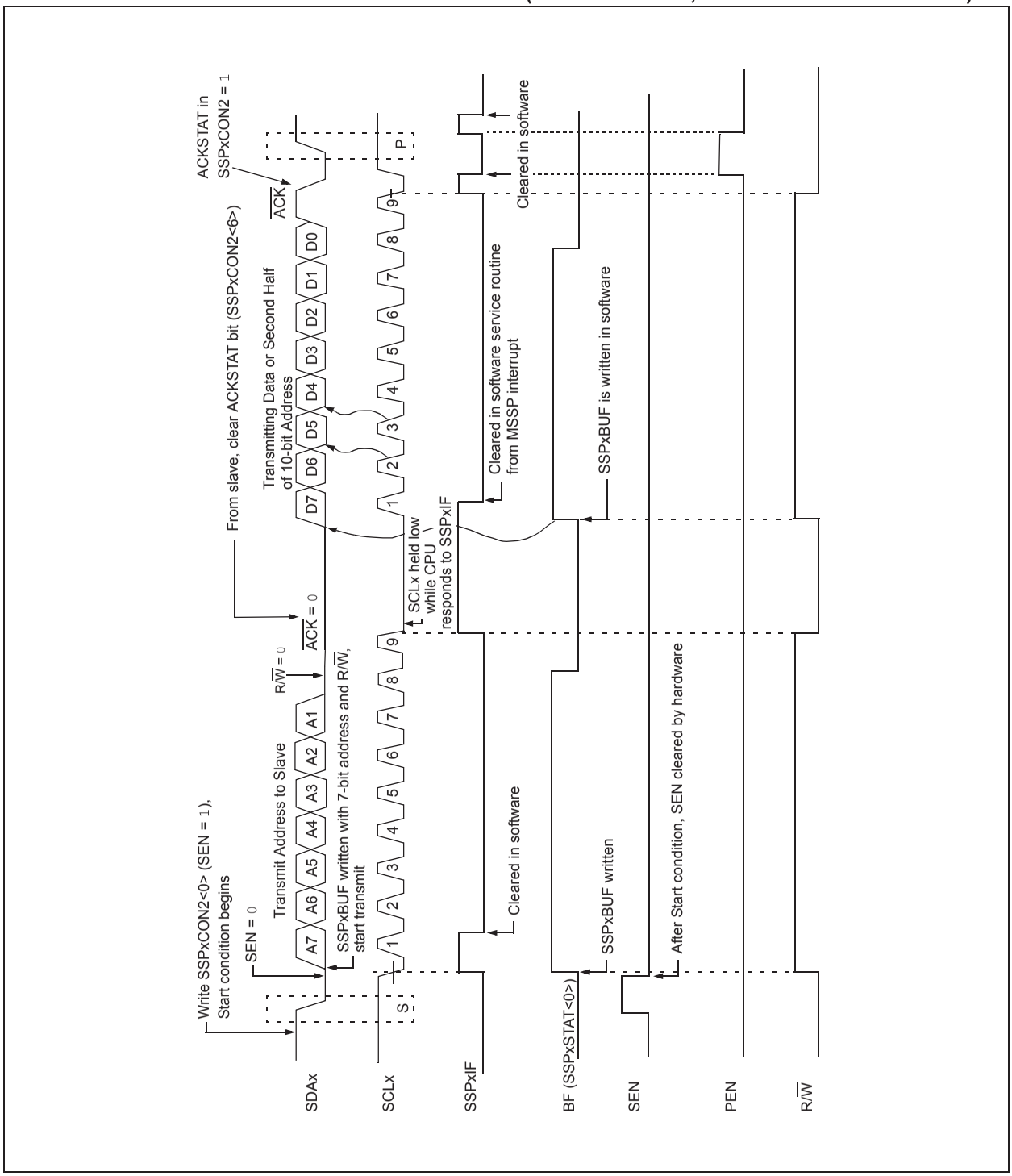


FIGURE 20-23: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7-BIT OR 10-BIT ADDRESS)



21.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs and so on.

The Enhanced USART module implements additional features, including Automatic Baud Rate Detection and calibration, automatic wake-up on Sync Break reception, and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

All members of the PIC18F47J13 Family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on character reception
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/CCP9/PMA5/TX1/CK1/RP17 and RC7/CCP10/PMA4/RX1/DT1/RP18), and remapped (RPn1/TX2/CK2 and RPn2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
  - SPEN bit (RCSTA1<7>) must be set (= 1)
  - TRISC<7> bit must be set (= 1)
  - TRISC<6> bit must be cleared (= 0) for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode
- For EUSART2:
  - SPEN bit (RCSTA2<7>) must be set (= 1)
  - TRIS bit for RPn2/RX2/DT2 = 1
  - TRIS bit for RPn1/TX2/CK2 = 0 for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode

Note:	The EUSART control will automatically reconfigure the pin from input to output as needed.

The TXx/CKx I/O pins have an optional open-drain output capability. By default, when this pin is used by the EUSART as an output, it will function as a standard push-pull CMOS output. The TXx/CKx I/O pins' open-drain, output feature can be enabled by setting the corresponding UxOD bit in the ODCON2 register. For more details, see **Section 20.3.3 "Open-Drain Output Option"**.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are covered in detail in Register 21-1, Register 21-2 and Register 21-3, respectively.

Note:	Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

22.0 10/12-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module in the PIC18F47J13 family of devices has 10 inputs for the 28-pin devices and 13 inputs for the 44-pin devices. This module allows conversion of an analog input signal to a corresponding 10- or 12-bit digital number.

The module has these registers:

- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Port Configuration Register 0 (ANCON0)
- A/D Port Configuration Register 1 (ANCON1)
- A/D Result Registers (ADRESH and ADRESL)
- A/D Trigger Register (ADCTRIG)
- Configuration Register 3 High (ADCSEL, CONFIG3H<1>)

The ADCON0 register, shown in Register 22-1, controls the operation of the A/D module.

The ADCON1 register, shown in Register 22-2, configures the A/D clock source, programmed acquisition time and justification. The ANCON0 and ANCON1 registers, in Register 22-1 and Register 22-2, configure the functions of the port pins.

The ADCSEL Configuration bit (CONFIG3H<1>) sets the module for 10- or 12-bit conversions. The 10-Bit Conversion mode is useful for applications that favor conversion speed over conversion resolution.

23.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake-up the device from Sleep mode when enabled. Each operational comparator will consume additional current. To minimize power consumption while in Sleep mode, turn off the comparators (CON = 0) before entering Sleep. If the device wakes up from Sleep, the contents of the CMxCON register are not affected.

23.8 Effects of a Reset

A device Reset forces the CMxCON registers to their Reset state. This forces both comparators and the voltage reference to the OFF state.

TABLE 23-3: REGISTERS ASSOCIATED WITH COMPARATOR 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
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR5	—	—	CM3IF	TMR8IF	TMR6IF	TMR5IF	TMR5GIF	TMR1GIF
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE5	—	—	CM3IE	TMR8IE	TMR6IE	TMR5IE	TMR5GIE	TMR1GIE
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR5	—	—	CM3IP	TMR8IP	TMR6IP	TMR5IP	TMR5GIP	TMR1GIP
CMxCON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
CMSTAT	—	—	—	—	—	COUT3	COUT2	COUT1
ANCON0	PCFG7(1)	PCFG6(1)	PCFG5(1)	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
ANCON1	VBGEN	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0

Legend: — = unimplemented, read as '0'. Shaded cells are not related to comparator operation.

Note 1: These bits are not implemented on 28-pin devices.

30.2 DC Characteristics: Power-Down and Supply Current PIC18F47J13 Family (Industrial) (Continued)

PIC18LF47J13 Family		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F47J13 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	PIC18LFXXJ13	9	45	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 32 kHz(3), **SEC_RUN** mode, SOSCSEL = 0b01
		9	45	µA	+25°C		
		12	61	µA	+85°C		
	PIC18FXXJ13	24	95	µA	-40°C	VDD = 2.15V, VDDCORE = 10 µF	
		28	95	µA	+25°C		
		35	105	µA	+85°C		
	PIC18FXXJ13	27	110	µA	-40°C	VDD = 3.3V, VDDCORE = 10 µF	
		31	110	µA	+25°C		
		35	150	µA	+85°C		
	PIC18LFXXJ13	2.5	31	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 32 kHz(3), **SEC_IDLE** mode, SOSCSEL = 0b01
		3.0	31	µA	+25°C		
		6.1	50	µA	+85°C		
	PIC18FXXJ13	19	87	µA	-40°C	VDD = 2.15V, VDDCORE = 10 µF	
		24	89	µA	+25°C		
		31	97	µA	+85°C		
	PIC18FXXJ13	21	100	µA	-40°C	VDD = 3.3V, VDDCORE = 10 µF	
		25	100	µA	+25°C		
		31	140	µA	+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 a high-impedance state and tied to VDD or VSS, and all features that add delta current are 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 (such as WDT). 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/VSS;
$\overline{MCLR}$ = VDD; WDT disabled unless otherwise specified.

3: Low-power Timer1 with 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.

30.5.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 30-5: EXTERNAL CLOCK TIMING

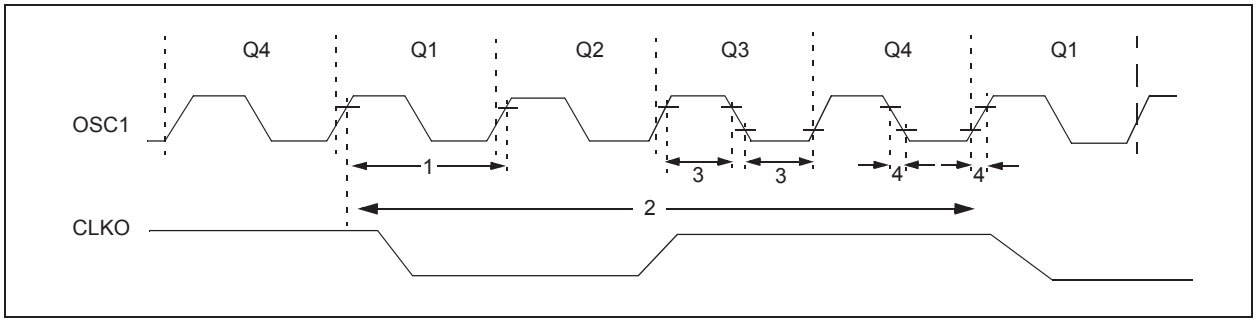


TABLE 30-9: EXTERNAL CLOCK TIMING REQUIREMENTS

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	16	MHz	HS Oscillator mode
			4	16(4)		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)	62.5	250	ns	HS Oscillator mode
			62.5(4)	250		HSPLL Oscillator mode(3)
2	TCY	Instruction Cycle Time(1)	83.3	DC	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.

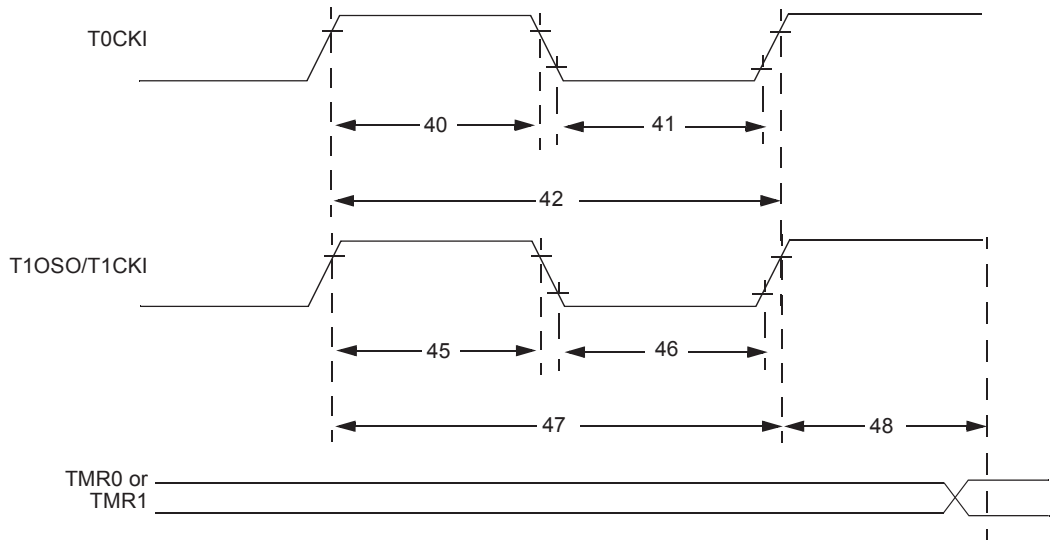
3: In order to use the PLL, the crystal/resonator must produce a frequency of either 4, 8, 12 or 16 MHz.

4: This is the maximum crystal/resonator driver frequency. The internal FOSC frequency, when running from the PLL, can be up to 48 MHz.

TABLE 30-15: LOW-POWER WAKE-UP TIME

Param. No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
W1	WDS	Deep Sleep	—	500	—	µs	REGSLP = 1
W2	WSLEEP	Sleep	—	35	—	µs	REGSLP = 1, PLLEN = 0, FOSC = 8 MHz INTOSC
W3	WDOZE1	Sleep	—	12	—	µs	REGSLP = 0, PLLEN = 0, FOSC = 8 MHz INTOSC
W4	WDOZE2	Sleep	—	1.1	—	µs	REGSLP = 0, PLLEN = 0, FOSC = 8 MHz EC
W5	WDOZE3	Sleep	—	230	—	ns	REGSLP = 0, PLLEN = 0, FOSC = 48 MHz EC
W6	WIDLE	Idle	—	230	—	ns	FOSC = 48 MHz EC

FIGURE 30-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Note: Refer to Figure 30-4 for load conditions.

FIGURE 30-12: PARALLEL SLAVE PORT TIMING

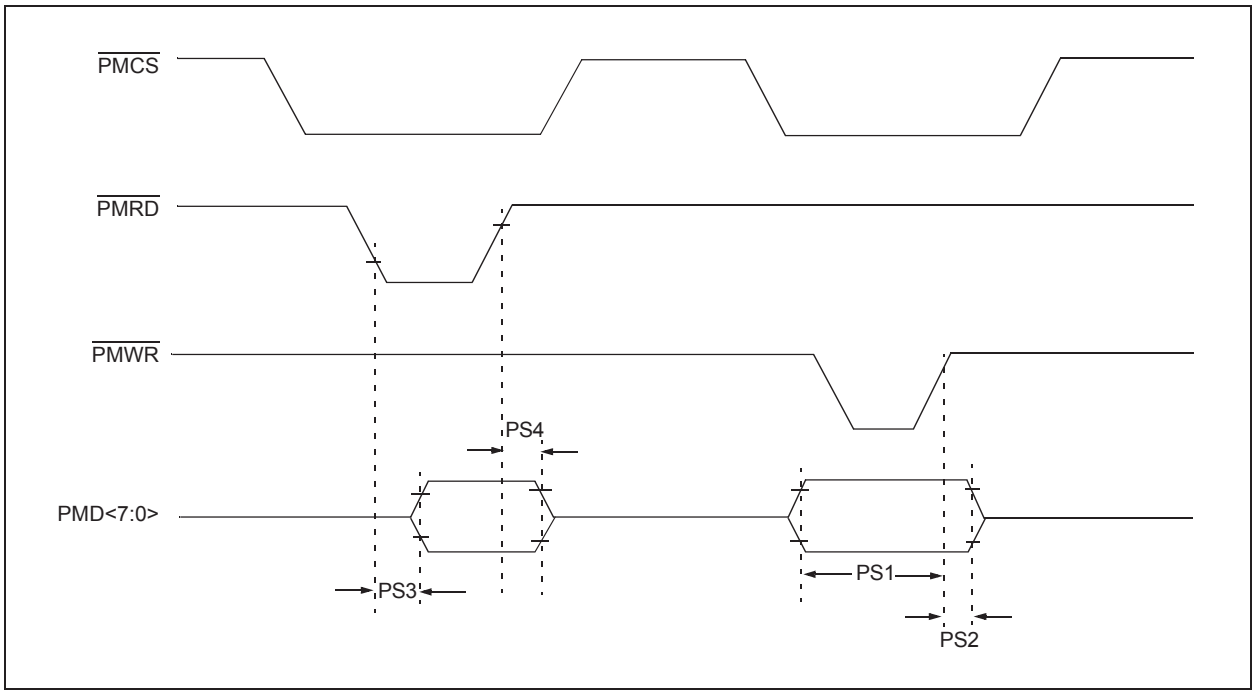
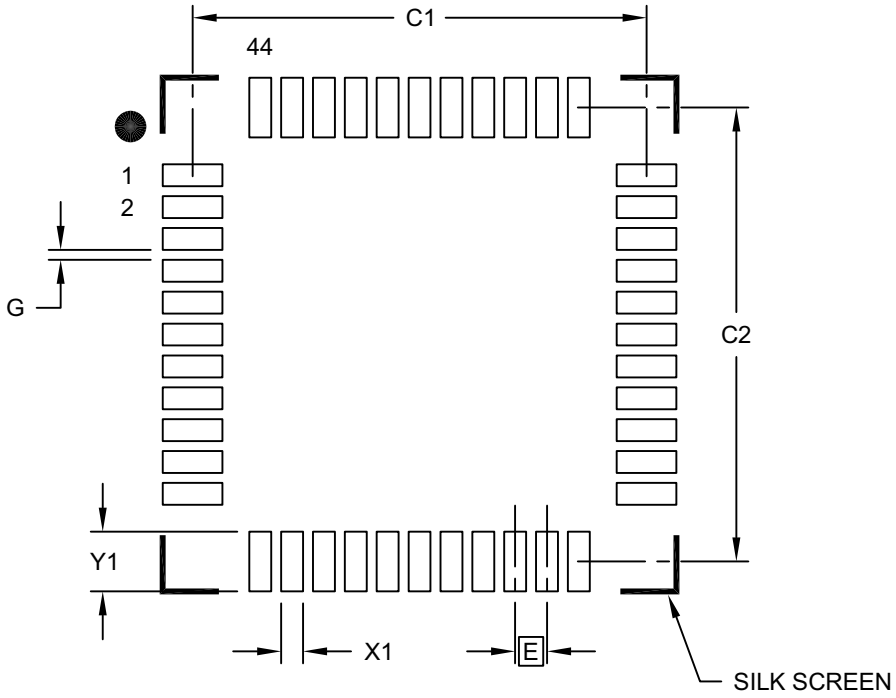


TABLE 30-20: PARALLEL SLAVE PORT REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
PS1	TdtV2wrH	Data In Valid before $\overline{\text{PMWR}}$ or $\overline{\text{PMCS}}$ Inactive (setup time)	20	—	—	ns	
PS2	TwrH2dtI	$\overline{\text{PMWR}}$ or $\overline{\text{PMCS}}$ Inactive to Data–In Invalid (hold time)	20	—	—	ns	
PS3	TrdL2dtV	$\overline{\text{PMRD}}$ and $\overline{\text{PMCS}}$ Active to Data–Out Valid	—	—	80	ns	
PS4	TrdH2dtI	$\overline{\text{PMRD}}$ Inactive or $\overline{\text{PMCS}}$ Inactive to Data–Out Invalid	10	—	30	ns	

44-Lead Plastic Thin Quad Flatpack (PT) - 10X10X1 mm Body, 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.80 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B