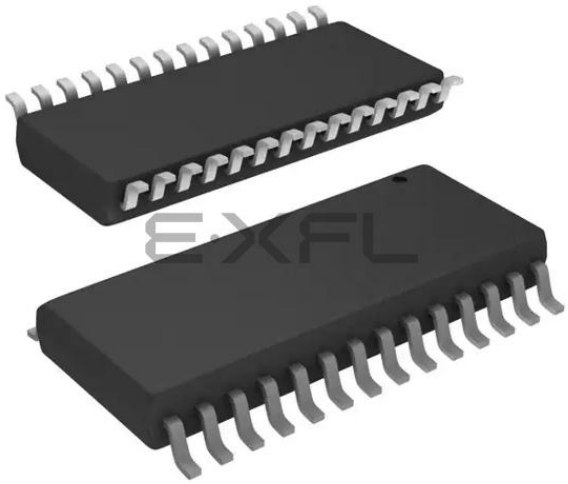


Welcome to E-XFL.COM

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.

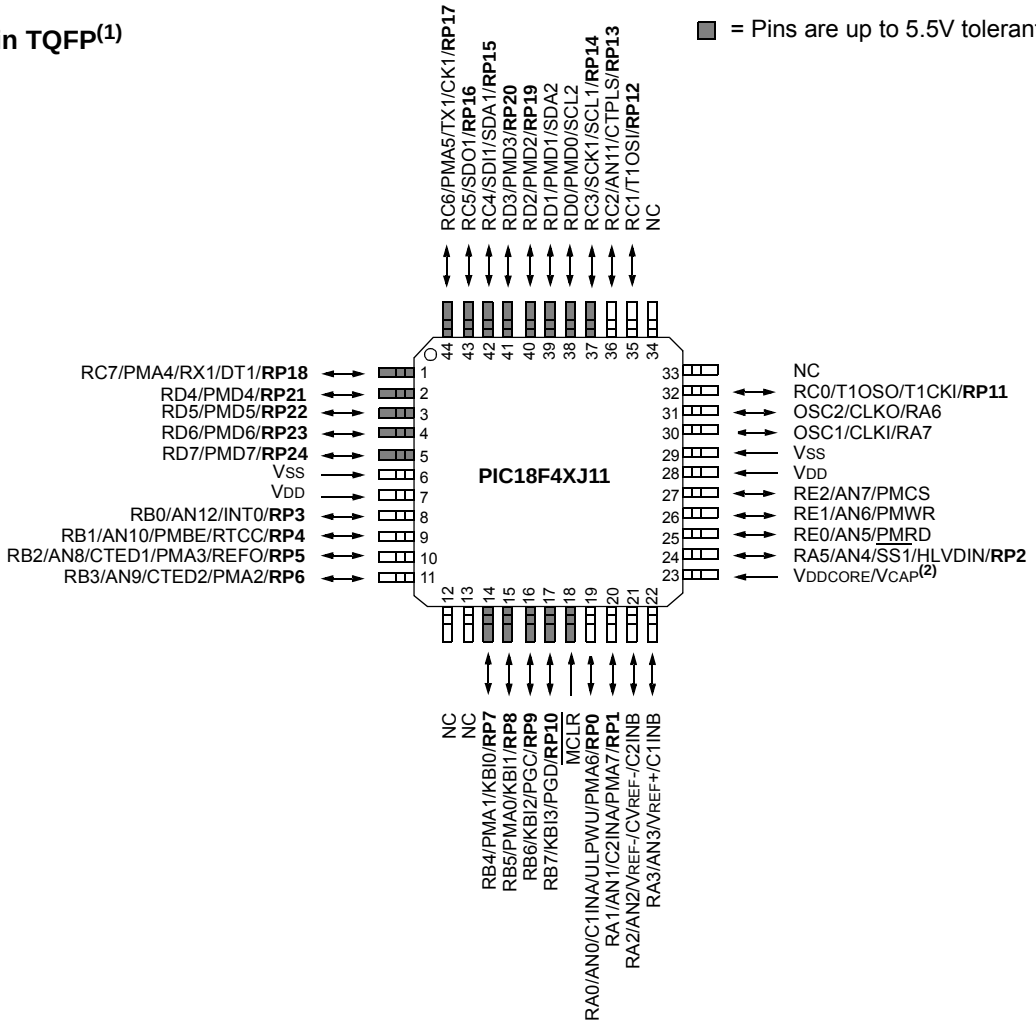
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
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 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26j11-i-so

Pin Diagrams (Continued)

44-Pin TQFP(1)

= Pins are up to 5.5V tolerant



Legend: **RPn** represents remappable pins.

Note 1: Some input and output functions are routed through the Peripheral Pin Select (PPS) module and can be dynamically assigned to any of the RPn pins. For a list of the input and output functions, see Table 10-13 and Table 10-14, respectively. For details on configuring the PPS module, see **Section 10.7 "Peripheral Pin Select (PPS)"**.

2: See **Section 26.3 "On-Chip Voltage Regulator"** for details on how to connect the VDDCORE/VCAP pin.

NOTES:

2.4 Voltage Regulator Pins (VCAP/ VDDCORE)

When the regulator is enabled ("F" devices), a low-ESR (< 5Ω) capacitor is required on the VCAP/VDDCORE pin to stabilize the voltage regulator output voltage. The VCAP/ VDDCORE pin must not be connected to VDD and must use a capacitor of 10 µF connected to ground. The type can be ceramic or tantalum. Suitable examples of capacitors are shown in Table 2-1. Capacitors with equivalent specifications can be used.

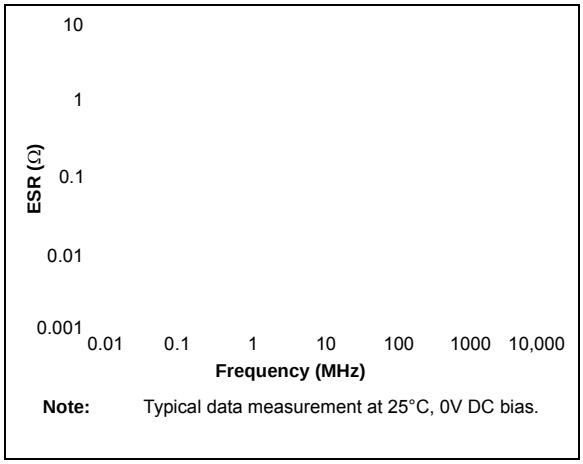
Designers may use Figure 2-3 to evaluate ESR equivalence of candidate devices.

It is recommended that the trace length not exceed 0.25 inch (6 mm). Refer to **Section 28.0 "Electrical Characteristics"** for additional information.

When the regulator is disabled ("LF" devices), the VCAP/VDDCORE pin must be tied to a voltage supply at the VDDCORE level. Refer to **Section 28.0 "Electrical Characteristics"** for information on VDD and VDDCORE.

Note that the "LF" versions are provided with the voltage regulator permanently disabled; they must always be provided with a supply voltage on the VDDCORE pin.

FIGURE 2-3: FREQUENCY vs. ESR PERFORMANCE FOR SUGGESTED VCAP



Note: Typical data measurement at 25°C, 0V DC bias.

TABLE 2-1: SUITABLE CAPACITOR EQUIVALENTS

Make	Part #	Nominal Capacitance	Base Tolerance	Rated Voltage	Temp. Range
TDK	C3216X7R1C106K	10 µF	±10%	16V	-55 to 125ºC
TDK	C3216X5R1C106K	10 µF	±10%	16V	-55 to 85ºC
Panasonic	ECJ-3YX1C106K	10 µF	±10%	16V	-55 to 125ºC
Panasonic	ECJ-4YB1C106K	10 µF	±10%	16V	-55 to 85ºC
Murata	GRM32DR71C106KA01L	10 µF	±10%	16V	-55 to 125ºC
Murata	GRM31CR61C106KC31L	10 µF	±10%	16V	-55 to 85ºC

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
PMDOUT2H(5)	—	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PMDOUT2L(5)	—	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PMDIN2H(5)	—	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PMDIN2L(5)	—	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PMEH(5)	—	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PMEL(5)	—	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
PMSTATH(5)	—	PIC18F4XJ11	00-- 0000	00-- 0000	uu-- uuuu
PMSTATL(5)	—	PIC18F4XJ11	10-- 1111	10-- 1111	uu-- uuuu
CVRCON	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
TCLKCON	PIC18F2XJ11	PIC18F4XJ11	---0 --00	---0 --uu	---u --uu
DSGPR1(6)	PIC18F2XJ11	PIC18F4XJ11	uuuu uuuu	uuuu uuuu	uuuu uuuu
DSGPR0(6)	PIC18F2XJ11	PIC18F4XJ11	uuuu uuuu	uuuu uuuu	uuuu uuuu
DSCONH(6)	PIC18F2XJ11	PIC18F4XJ11	0--- -000	0--- -uuu	u--- -uuu
DSCONL(6)	PIC18F2XJ11	PIC18F4XJ11	---- -000	---- -u00	---- -uuu
DSWAKEH(6)	PIC18F2XJ11	PIC18F4XJ11	---- ---0	---- ---0	---- ---u
DSWAKEL(6)	PIC18F2XJ11	PIC18F4XJ11	0-00 00-1	0-00 00-0	u-uu uu-u
ANCON1	PIC18F2XJ11	PIC18F4XJ11	00-0 0000	00-0 0000	uu-u uuuu
ANCON0	PIC18F2XJ11	PIC18F4XJ11	0000 0000	0000 0000	uuuu uuuu
ODCON1	PIC18F2XJ11	PIC18F4XJ11	---- --00	---- --uu	---- --uu
ODCON2	PIC18F2XJ11	PIC18F4XJ11	---- --00	---- --uu	---- --uu
ODCON3	PIC18F2XJ11	PIC18F4XJ11	---- --00	---- --uu	---- --uu
RTCCFG	PIC18F2XJ11	PIC18F4XJ11	0-00 0000	u-uu uuuu	u-uu uuuu
RTCCAL	PIC18F2XJ11	PIC18F4XJ11	0000 0000	uuuu uuuu	uuuu uuuu
REFOCON	PIC18F2XJ11	PIC18F4XJ11	0-00 0000	0-00 0000	u-uu uuuu
PADCFG1	PIC18F2XJ11	PIC18F4XJ11	---- -000	---- -000	---- -uuu
PPSCON	PIC18F2XJ11	PIC18F4XJ11	---- ---0	---- ---0	---- ---u
RPINR24	PIC18F2XJ11	PIC18F4XJ11	---1 1111	---1 1111	---u uuuu
RPINR23	PIC18F2XJ11	PIC18F4XJ11	---1 1111	---1 1111	---u uuuu
RPINR22	PIC18F2XJ11	PIC18F4XJ11	---1 1111	---1 1111	---u uuuu
RPINR21	PIC18F2XJ11	PIC18F4XJ11	---1 1111	---1 1111	---u uuuu
RPINR17	PIC18F2XJ11	PIC18F4XJ11	---1 1111	---1 1111	---u uuuu
RPINR16	PIC18F2XJ11	PIC18F4XJ11	---1 1111	---1 1111	---u 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 PIC18F2XJ11 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 and Table 6-3 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 EC0h 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	RTCVALH	F79h	T3CON
FF8h	TBLPTRU	FD8h	STATUS	FB8h	ECCP2AS	F98h	RTCVALL	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	ALRMCFG	F71h	SSP2CON2
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	ALRMRPT	F70h	CMSTAT
FEFh	INDF0(1)	FCFh	TMR1H	FAFh	RCREG1	F8Fh	ALRMVALH	F6Fh	PMADDRH(2,4)
FEEh	POSTINC0(1)	FCEh	TMR1L	FAEh	TXREG1	F8Eh	ALRMVALL	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	—(5)	F67h	DMABCL
FE6h	POSTINC1(1)	FC6h	SSP1CON1	FA6h	EECON1	F86h	DMACON2	F66h	DMABCH
FE5h	POSTDEC1(1)	FC5h	SSP1CON2	FA5h	IPR3	F85h	HLVDCON	F65h	—(5)
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE(2)	F64h	—(5)
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD(2)	F63h	—(5)
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	—(5)
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	—(5)
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	—(5)

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.

7.4 Erasing Flash Program Memory

The minimum erase block is 512 words or 1024 bytes. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the micro-controller itself, a block of 1024 bytes of program memory is erased. The Most Significant 12 bits of the TBLPTR<21:10> point to the block being erased. TBLPTR<9:0> are ignored.

The EECON1 register commands the erase operation. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. For protection, the write initiate sequence for EECON2 must be used.

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

7.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load Table Pointer register with address of row being erased.
2. Set the WREN and FREE bits (EECON1<2,4>) to enable the erase operation.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit; this will begin the erase cycle.
7. The CPU will stall for the duration of the erase for TIE (see parameter D133B).
8. Re-enable interrupts.

EXAMPLE 7-2: ERASING FLASH PROGRAM MEMORY

```
                MOVLW   CODE_ADDR_UPPER     ; load TBLPTR with the base
                MOVWF   TBLPTRU             ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
    ERASE_ROW
                BSF     EECON1, WREN        ; enable write to memory
                BSF     EECON1, FREE        ; enable Erase operation
                BCF     INTCON, GIE         ; disable interrupts
Required        MOVLW   0x55
Sequence        MOVWF   EECON2              ; write 55h
                MOVLW   0xAA
                MOVWF   EECON2              ; write 0AAh
                BSF     EECON1, WR          ; start erase (CPU stall)
                BSF     INTCON, GIE         ; re-enable interrupts
```

9.2 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Request (Flag) registers (PIR1, PIR2, PIR3).

Note 1: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON<7>).

2: User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1 (ACCESS F9Eh)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIF(1)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
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 **PMPIF:** Parallel Master Port Read/Write Interrupt Flag bit(1)
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5 **RC1IF:** EUSART1 Receive Interrupt Flag bit
1 = The EUSART1 receive buffer, RCREG1, is full (cleared when RCREG1 is read)
0 = The EUSART1 receive buffer is empty

bit 4 **TX1IF:** EUSART1 Transmit Interrupt Flag bit
1 = The EUSART1 transmit buffer, TXREG1, is empty (cleared when TXREG1 is written)
0 = The EUSART1 transmit buffer is full

bit 3 **SSP1IF:** Master Synchronous Serial Port 1 Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2 **CCP1IF:** ECCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
0 = No TMR1/TMR3 register capture occurred
Compare mode:
1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1/TMR3 register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Note 1: These bits are unimplemented on 28-pin devices.

9.6 INTx Pin Interrupts

External interrupts on the INT0, INT1, INT2 and INT3 pins are edge-triggered. If the corresponding INTEDGx bit in the INTCON2 register is set (= 1), the interrupt is triggered by a rising edge; if the bit is clear, the trigger is on the falling edge. When a valid edge appears on the INTx pin, the corresponding flag bit and INTxIF are set. This interrupt can be disabled by clearing the corresponding enable bit, INTxIE. Flag bit, INTxIF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from the Sleep and Idle modes if bit, INTxIE, was set prior to going into the power-managed modes. After waking from Sleep or Idle mode, the processor will branch to the interrupt vector if the Global Interrupt Enable bit (GIE) is set. Deep Sleep mode can wake up from INT0, but the processor will start execution from the Power-on Reset vector rather than branch to the interrupt vector.

Interrupt priority for INT1, INT2 and INT3 is determined by the value contained in the Interrupt Priority bits, INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0. It is always a high-priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (FFh → 00h) will set flag bit, TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh → 0000h) will set TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 12.0 "Timer0 Module"** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit, RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

9.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the Fast Return Stack. If a fast return from interrupt is not used (see **Section 6.3 "Data Memory Organization"**), the user may need to save the WREG, STATUS and BSR registers on entry to the Interrupt Service Routine. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine.

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF   W_TEMP                  ; W_TEMP is in access bank
MOVFF   STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF   BSR, BSR_TEMP           ; BSR_TEMP located anywhere
;
; USER ISR CODE
;
MOVFF   BSR_TEMP, BSR           ; Restore BSR
MOVF    W_TEMP, W               ; Restore WREG
MOVFF   STATUS_TEMP, STATUS     ; Restore STATUS
```

REGISTER 10-24: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3 (BANKED EC9h)

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
—	—	—	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable 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 **RP3R<4:0>:** Peripheral Output Function is Assigned to RP3 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-25: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4 (BANKED ECAh)

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
—	—	—	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable 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 **RP4R<4:0>:** Peripheral Output Function is Assigned to RP4 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-26: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5 (BANKED ECBh)

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
—	—	—	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable 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 **RP5R<4:0>:** Peripheral Output Function is Assigned to RP5 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 11-5: PMEH: PARALLEL PORT ENABLE REGISTER HIGH BYTE (BANKED F57h)(1)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	PTEN14	—	—	—	—	—	—
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 **Unimplemented**: Read as '0'

bit 6 **PTEN14:** PMCS Port Enable bit
1 = PMCS chip select line
0 = PMCS functions as port I/O

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

Note 1: This register is only available in 44-pin devices.

REGISTER 11-6: PMEL: PARALLEL PORT ENABLE REGISTER LOW BYTE (BANKED F56h)(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
PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
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 7-2 **PTEN<7:2>:** PMP Address Port Enable bits
1 = PMA<7:2> function as PMP address lines
0 = PMA<7:2> function as port I/O

bit 1-0 **PTEN<1:0>:** PMALH/PMALL Strobe Enable bits
1 = PMA<1:0> function as either PMA<1:0> or PMALH and PMALL
0 = PMA<1:0> pads functions as port I/O

Note 1: This register is only available in 44-pin devices.

FIGURE 11-24: WRITE TIMING, 16-BIT MULTIPLEXED DATA, PARTIALLY MULTIPLEXED ADDRESS

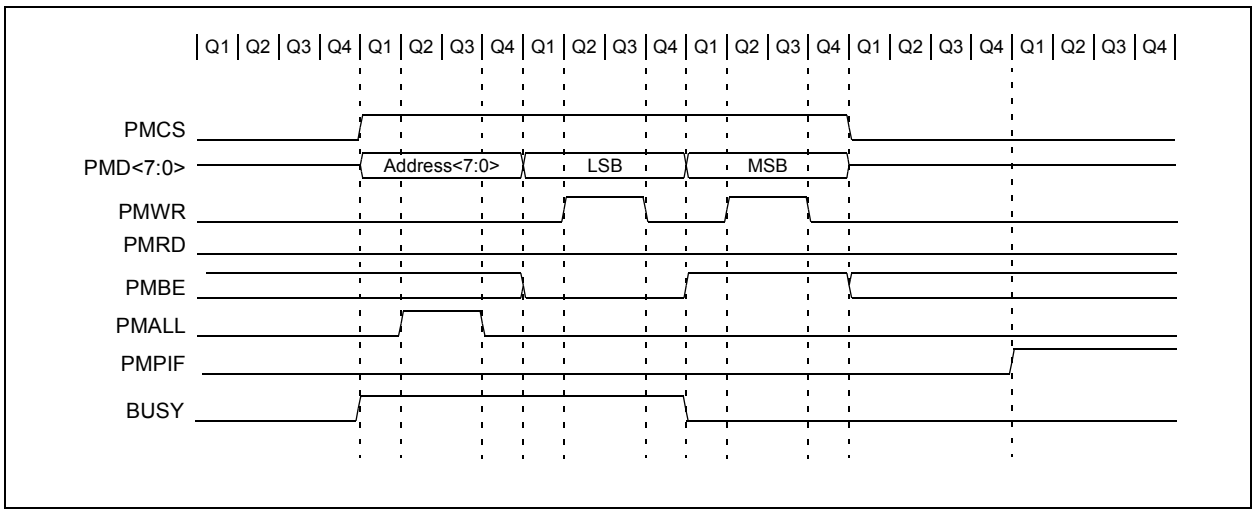


FIGURE 11-25: READ TIMING, 16-BIT MULTIPLEXED DATA, FULLY MULTIPLEXED 16-BIT ADDRESS

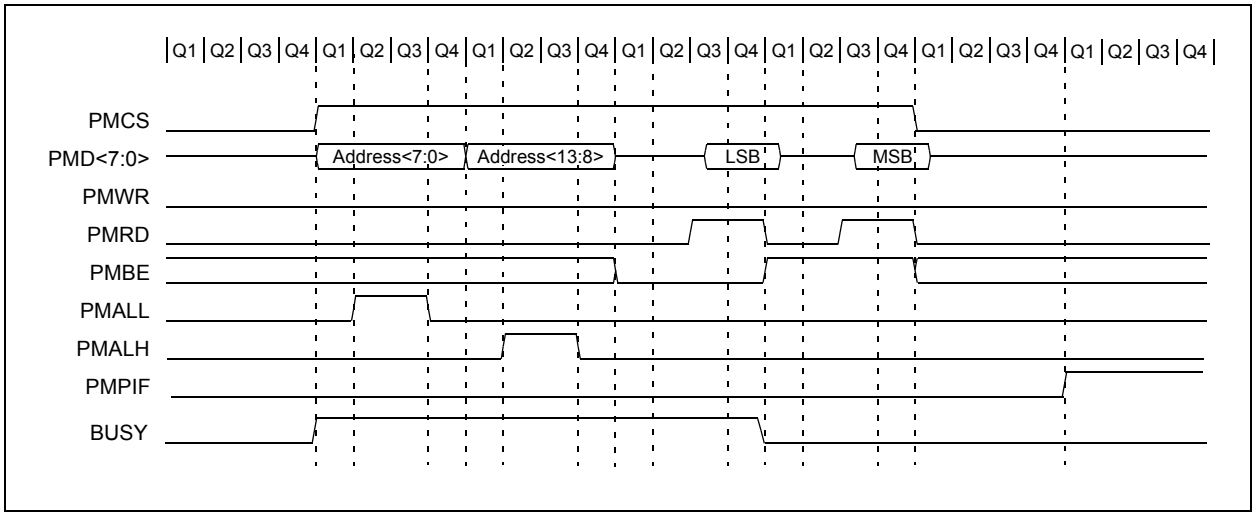
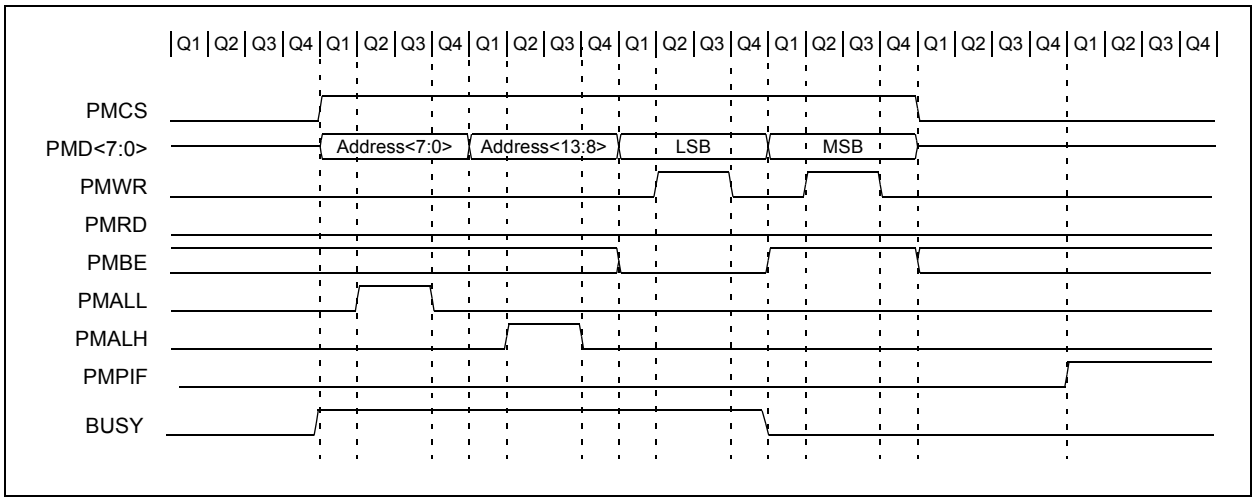


FIGURE 11-26: WRITE TIMING, 16-BIT MULTIPLEXED DATA, FULLY MULTIPLEXED 16-BIT ADDRESS



13.2 Timer1 Operation

The Timer1 module is an 8-bit or 16-bit incrementing counter, which is accessed through the TMR1H:TMR1L register pair.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively.

When Timer1 is enabled, the RC1/T1OSI/RP12 and RC0/T1OSO/T1CKI/RP11 pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

13.3 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Register 13-1 displays the clock source selections.

When switching clock sources and using the clock prescaler, write to TMR1L afterwards to reset the internal prescaler count to 0.

13.3.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

13.3.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input, T1CKI, or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

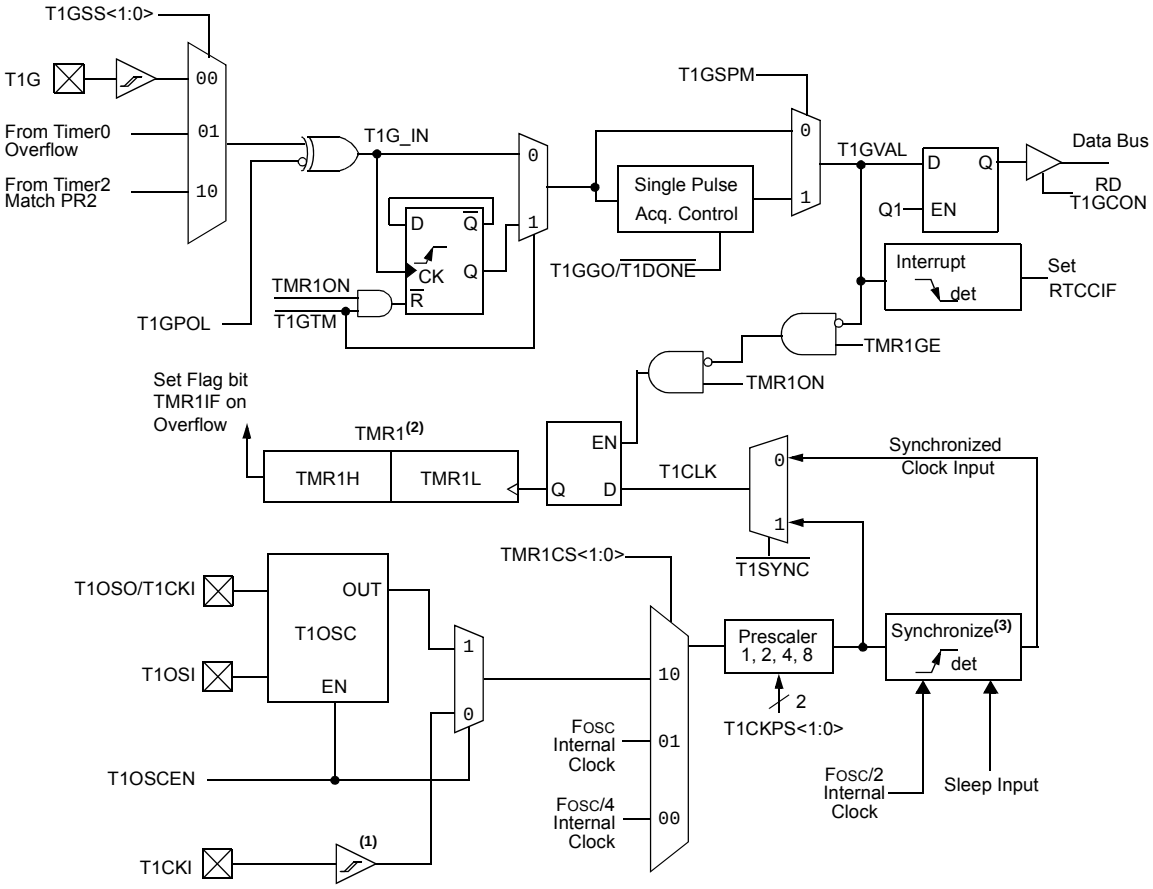
- Timer1 enabled after POR Reset
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0)

when T1CKI is high, then Timer1 is enabled (TMR1ON = 1) when T1CKI is low.

TABLE 13-1: TIMER1 CLOCK SOURCE SELECTION

TMR1CS1	TMR1CS0	T1OSCEN	Clock Source
0	1	x	Clock Source (FOSC)
0	0	x	Instruction Clock (FOSC/4)
1	0	0	External Clock on T1CKI Pin
1	0	1	Oscillator Circuit on T1OSI/T1OSO Pin

FIGURE 13-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high-speed type when using T1CKI.
2: Timer1 register increments on rising edge.
3: Synchronize does not operate while in Sleep.

19.3.1 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.

Note:	Because the SSPxBUF register is double-buffered, using read-modify-write instructions such as BCF, COMF, etc., will not work. Similarly, when debugging under an in-circuit debugger, performing actions that cause reads of SSPxBUF (mouse hovering, watch, etc.) can consume data that the application code was expecting to receive.

REGISTER 19-1: SSPxSTAT: MSSPx STATUS REGISTER – SPI MODE (ACCESS FC7h/F73h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
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/$\overline{Address}$ bit
Used in I2C™ mode only.

bit 4 **P:** Stop bit
Used in I2C mode only; this bit is cleared when the MSSP module is disabled, SSPEN is cleared.

bit 3 **S:** Start bit
Used in I2C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ Information bit
Used in I2C mode only.

bit 1 **UA:** Update Address bit
Used in I2C mode only.

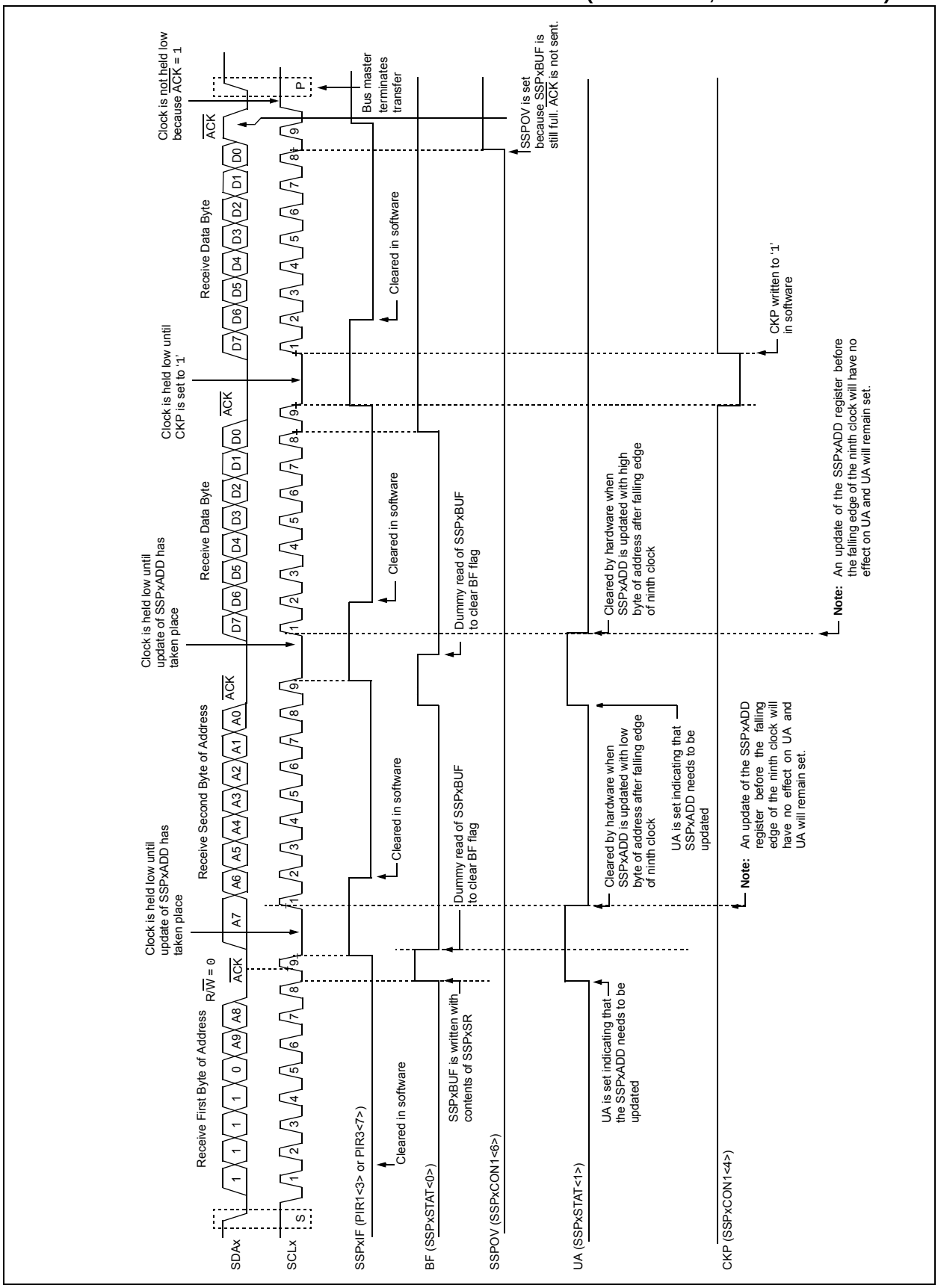
bit 0 **BF:** Buffer Full Status bit
1 = Receive complete, SSPxBUF is full
0 = Receive not complete, SSPxBUF is empty

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

EXAMPLE 19-2: 512-BYTE SPI MASTER MODE Init AND TRANSFER

```
                                    ;For this example, let's use RP5(RB2) for SCK2,
                                    ;RP4(RB1) for SDO2, and RP3(RB0) for SDI2

                                    ;Let's use SPI master mode, CKE = 0, CKP = 0,
                                    ;without using slave select signalling.

InitSPIPins:
    movlb       0x0F                ;Select bank 15, for access to ODCON3 register
    bcf         ODCON3, SPI2OD      ;Let's not use open drain outputs in this example

    bcf         LATB, RB2           ;Initialize our (to be) SCK2 pin low (idle).
    bcf         LATB, RB1           ;Initialize our (to be) SDO2 pin to an idle state
    bcf         TRISB, RB1          ;Make SDO2 output, and drive low
    bcf         TRISB, RB2          ;Make SCK2 output, and drive low (idle state)
    bsf         TRISB, RB0          ;SDI2 is an input, make sure it is tri-stated

                                    ;Now we should unlock the PPS registers, so we can
                                    ;assign the MSSP2 functions to our desired I/O pins.

    movlb       0x0E                ;Select bank 14 for access to PPS registers
    bcf         INTCON, GIE         ;I/O Pin unlock sequence will not work if CPU
                                    ;services an interrupt during the sequence
    movlw       0x55                ;Unlock sequence consists of writing 0x55
    movwf       EECON2              ;and 0xAA to the EECON2 register.
    movlw       0xAA
    movwf       EECON2
    bcf         PPSCON, IOLOCK      ;We may now write to RPINRx and RPORx registers
    bsf         INTCON, GIE         ;May now turn back on interrupts if desired

    movlw       0x03                ;0x0A is SCK2 output signal
    movwf       RPINR21             ;Assign the SDI2 function to pin RP3

    movlw       0x0A                ;Let's assign SCK2 output to pin RP4
    movwf       RPOR4               ;RPOR4 maps output signals to RP4 pin
    movlw       0x04                ;SCK2 also needs to be configured as an input on the
                                     same pin
   movwf        RPINR22             ;SCK2 input function taken from RP4 pin
   movlw        0x09                ;0x09 is SDO2 output
   movwf        RPOR5               ;Assign SDO2 output signal to the RP5 (RB2) pin
   bsf          PPSCON, IOLOCK      ;Lock the PPS registers to prevent changes
    movlb       0x0F                ;Done with PPS registers, bank 15 has other SFRs

InitMSSP2:
    clrf        SSP2STAT            ;CKE = 0, SMP = 0 (sampled at middle of bit)
    movlw       b'00000000'         ;CKP = 0, SPI Master mode, Fosc/4
    movwf       SSP2CON1            ;MSSP2 initialized
    bsf         SSP2CON1, SSPEN     ;Enable the MSSP2 module

InitSPIDMA:
    movlw       b'00111110'         ;Full duplex, RX/TXINC enabled, no SSCON
    movwf       DMACON1             ;DLYINTEN is set, so DLYCYC3:DLYCYC0 = 1111
    movlw       b'11110000'         ;Minimum delay between bytes, interrupt
    movwf       DMACON2             ;only once when the transaction is complete
```

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



REGISTER 26-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

R/WO-1	R/WO-1	R/WO-1	U-0	U-1	U-1	U-1	R/WO-1
$\overline{\text{DEBUG}}$	XINST	STVREN	—	—	—	—	WDTEN
bit 7							bit 0

Legend:			
R = Readable bit	WO = Write-Once bit	U = Unimplemented bit, read as ‘0’	
-n = Value at Reset	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Background debugger disabled; RB6 and RB7 configured as general purpose I/O pins
0 = Background debugger enabled; RB6 and RB7 are dedicated to In-Circuit Debug

bit 6 **XINST:** Extended Instruction Set Enable bit
1 = Instruction set extension and Indexed Addressing mode enabled
0 = Instruction set extension and Indexed Addressing mode disabled

bit 5 **STVREN**: Stack Overflow/Underflow Reset Enable bit
1 = Reset on stack overflow/underflow enabled
0 = Reset on stack overflow/underflow disabled

bit 4-1 **Unimplemented:** Read as ‘0’

bit 0 **WDTEN:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled (control is placed on SWDTEN bit)

TABLE 29-5: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	VRGOUT	Regulator Output Voltage	2.35	2.5	2.7	V	Regulator enabled, VDD = 3.0V
	CEFC	External Filter Capacitor Value(1)	5.4	10	18	µF	ESR < 3Ω recommended ESR < 5Ω required

Note 1: CEFC applies for PIC18F devices in the family. For PIC18LF devices in the family, there is no specific minimum or maximum capacitance for VDDCORE, although proper supply rail bypassing should still be used.

TABLE 29-6: ULPWU SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D100	IULP	**Ultra Low-Power Wake-up Current**	—	60	—	nA	Net of I/O leakage and current sink at 1.6V on pin, VDD = 3.3V See Application Note *AN879,* "*Using the Microchip Ultra Low-Power Wake-up Module*" (DS00879)

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

FIGURE 29-3: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

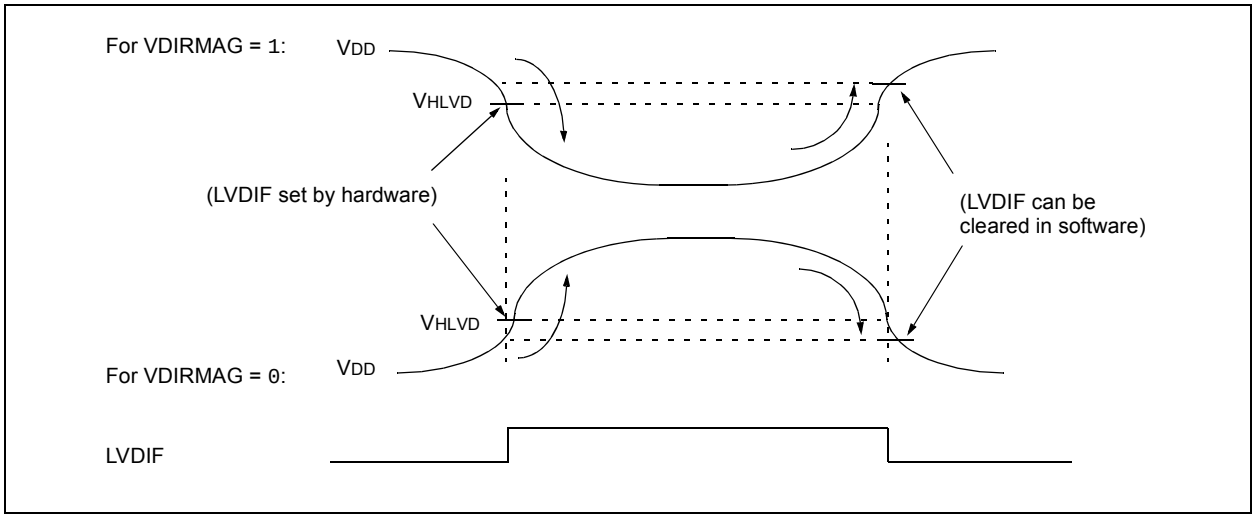
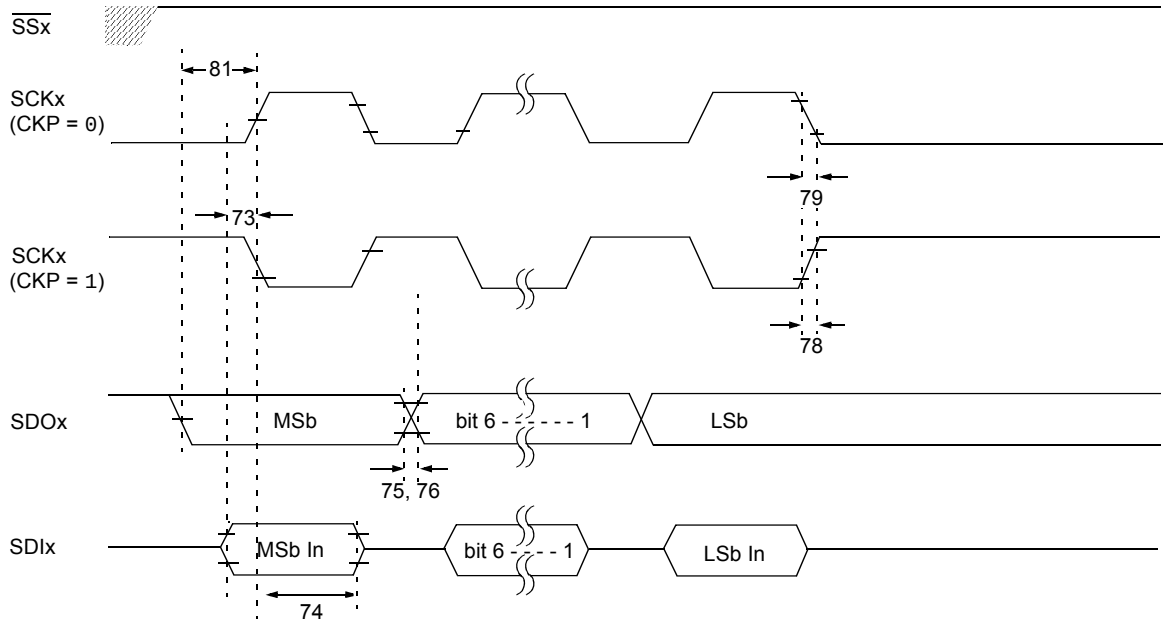


TABLE 29-7: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial								
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
D420		HLVD Voltage on VDD Transition High-to-Low	HLVDL<3:0> = 1000	2.33	2.45	2.57	V	
			HLVDL<3:0> = 1001	2.47	2.60	2.73	V	
			HLVDL<3:0> = 1010	2.66	2.80	2.94	V	
			HLVDL<3:0> = 1011	2.76	2.90	3.05	V	
			HLVDL<3:0> = 1100	2.85	3.00	3.15	V	
			HLVDL<3:0> = 1101	2.97	3.13	3.29	V	
			HLVDL<3:0> = 1110	3.23	3.40	3.57	V	
D421	TIRVST	Time for Internal Reference Voltage to become Stable		—	20	—	μs	
D422	TLVD	High/Low-Voltage Detect Pulse Width		200	—	—	μs	

FIGURE 29-14: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)



Note: Refer to Figure 29-4 for load conditions.

TABLE 29-21: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TDIV2SCH, TDIV2SCL	Setup Time of SDIx Data Input to SCKx Edge	35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			100	—	ns	VDD = 2.15V, VDDCORE = 2.15V
74	TSCH2DIL, TSCL2DIL	Hold Time of SDIx Data Input to SCKx Edge	30	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			83	—	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
78	TSCR	SCKx Output Rise Time (Master mode)	—	25	ns	PORTB or PORTC
79	TSCF	SCKx Output Fall Time (Master mode)	—	25	ns	PORTB or PORTC
81	TDOV2SCH, TDOV2SCL	SDOx Data Output Setup to SCKx Edge	TCY	—	ns	