Welcome to E-XFL.COM

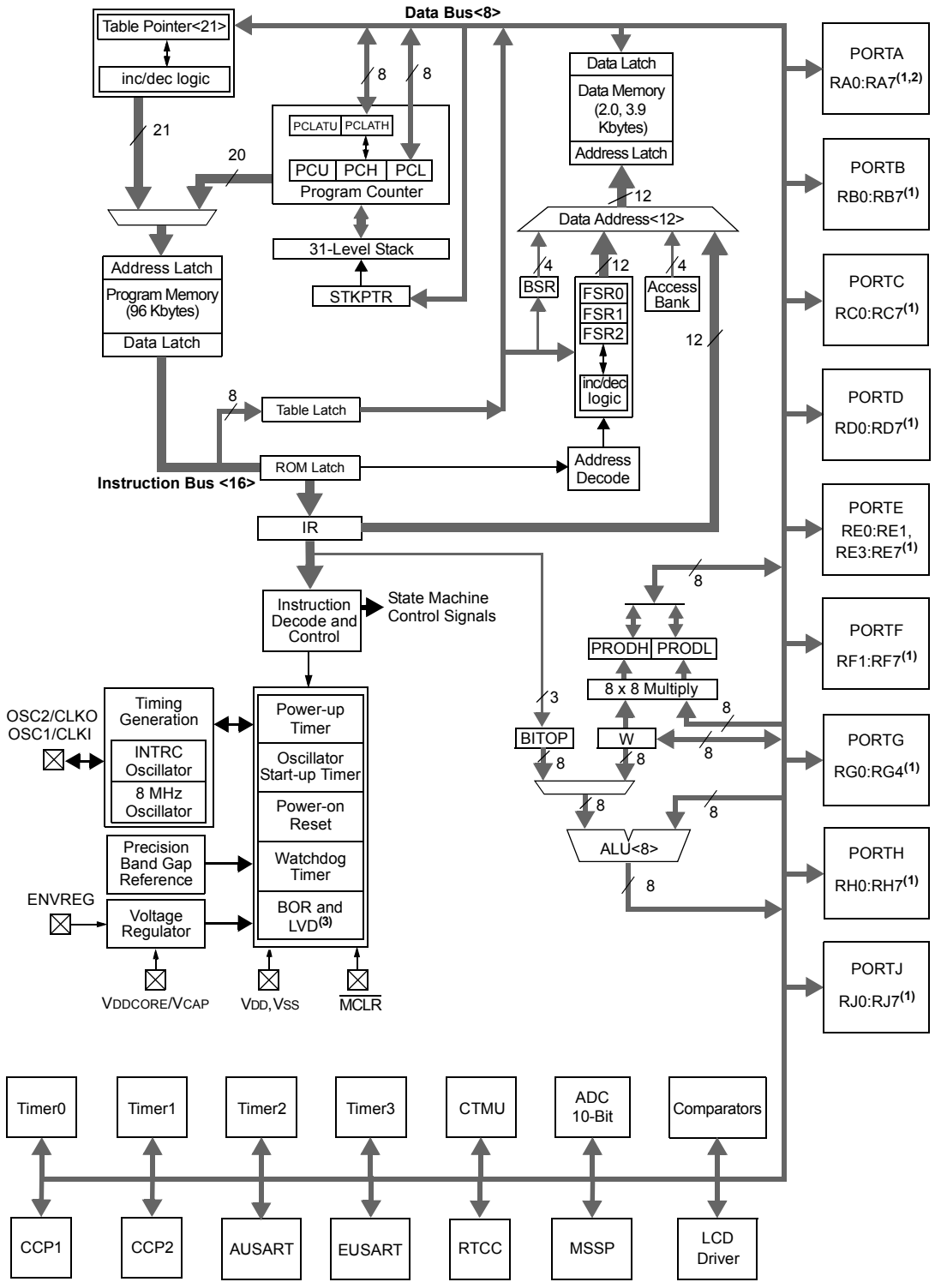
What is "Embedded - Microcontrollers"?

"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, LCD, LVD, POR, PWM, WDT
Number of I/O	67
Program Memory Size	128KB (64K 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 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f87j90-i-pt

FIGURE 1-2: PIC18F8XJ90 (80-PIN) BLOCK DIAGRAM



Note 1: See Table 1-3 for I/O port pin descriptions.

2: RA6 and RA7 are only available as digital I/O in select oscillator modes. See **Section 3.0 "Oscillator Configurations"** for more information.

3: Brown-out Reset and Low-Voltage Detect functions are provided when the on-board voltage regulator is enabled.

TABLE 1-3: PIC18F6XJ90 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTG is a bidirectional I/O port.
RG0/LCDBIAS0	3			
RG0		I/O	ST	Digital I/O.
LCDBIAS0		I	Analog	BIAS0 input for LCD.
RG1/TX2/CK2	4			
RG1		I/O	ST	Digital I/O.
TX2		O	—	AUSART asynchronous transmit.
CK2		I/O	ST	AUSART synchronous clock (see related RX2/DT2).
RG2/RX2/DT2/VLCAP1	5			
RG2		I/O	ST	Digital I/O.
RX2		I	ST	AUSART asynchronous receive.
DT2		I/O	ST	AUSART synchronous data (see related TX2/CK2).
VLCAP1		I	Analog	LCD charge pump capacitor input.
RG3/VLCAP2	6			
RG3		I/O	ST	Digital I/O.
VLCAP2		I	Analog	LCD charge pump capacitor input.
RG4/SEG26/RTCC	8			
RG4		I/O	ST	Digital I/O.
SEG26		O	Analog	SEG26 output for LCD.
RTCC		O	—	RTCC output
VSS	9, 25, 41, 56	P	—	Ground reference for logic and I/O pins.
VDD	26, 38, 57	P	—	Positive supply for logic and I/O pins.
AVSS	20	P	—	Ground reference for analog modules.
AVDD	19	P	—	Positive supply for analog modules.
ENVREG	18	I	ST	Enable for on-chip voltage regulator.
VDDCORE/VCAP	10			Core logic power or external filter capacitor connection.
VDDCORE		P	—	Positive supply for microcontroller core logic (regulator disabled).
VCAP		P	—	External filter capacitor connection (regulator enabled).

Legend: TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels; I = Input; P = Power; I^2C™ = I^2C/SMBus; CMOS = CMOS compatible input or output; Analog = Analog input; O = Output; OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for CCP2 when the CCP2MX Configuration bit is set.

2: Alternate assignment for CCP2 when the CCP2MX Configuration bit is cleared.

7.5.2 FLASH PROGRAM MEMORY WRITE SEQUENCE (WORD PROGRAMMING).

The PIC18F87J90 family of devices has a feature that allows programming a single word (two bytes). This feature is enabled when the WPROG bit is set. If the memory location is already erased, the following sequence is required to enable this feature:

1. Load the Table Pointer register with the address of the data to be written
2. Write the 2 bytes into the holding registers and perform a table write
3. Set WPROG to enable single-word write.
4. Set WREN to enable write to memory.
5. Disable interrupts.
6. Write 55h to EECON2.
7. Write 0AAh to EECON2.
8. Set the WR bit. This will begin the write cycle.
9. The CPU will stall for the duration of the write for TIW (see parameter D133A).
10. Re-enable interrupts.

EXAMPLE 7-4: SINGLE-WORD WRITE TO FLASH PROGRAM MEMORY

```
                MOVLW    CODE_ADDR_UPPER      ; Load TBLPTR with the base address
                MOVWF    TBLPTRU
                MOVLW    CODE_ADDR_HIGH
                MOVWF    TBLPTRH
                MOVLW    CODE_ADDR_LOW
                MOVWF    TBLPTRL

                MOVLW    DATA0
                MOVWF    TABLAT
                TBLWT*+
                MOVLW    DATA1
                MOVWF    TABLAT
                TBLWT*


PROGRAM_MEMORY
                BSF     EECON1, WPROG        ; enable single word write
                BSF     EECON1, WREN         ; enable write to memory
                BCF     INTCON, GIE          ; disable interrupts
                MOVLW   55h
    Required    MOVWF   EECON2               ; write 55h
    Sequence    MOVLW   0AAh
                MOVWF   EECON2               ; write 0AAh
                BSF     EECON1, WR           ; start program (CPU stall)
                BSF     INTCON, GIE          ; re-enable interrupts
                BCF     EECON1, WPROG        ; disable single word write
                BCF     EECON1, WREN         ; disable write to memory
```

Example 8-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES3:RES0).

EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L)
\end{aligned}
$$

EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 8-4 shows the sequence to do a 16 x 16 signed multiply. Equation 8-2 shows the algorithm used. The 32-bit result is stored in four registers (RES3:RES0). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L) + \\
&\quad (-1 \bullet ARG2H\langle 7\rangle \bullet ARG1H:ARG1L \bullet 2^{16}) + \\
&\quad (-1 \bullet ARG1H\langle 7\rangle \bullet ARG2H:ARG2L \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
   MOVF    ARG1L, W
   MULWF   ARG2L           ; ARG1L * ARG2L ->
                           ; PRODH:PRODL
   MOVFF   PRODH, RES1     ;
   MOVFF   PRODL, RES0     ;
;
   MOVF    ARG1H, W
   MULWF   ARG2H           ; ARG1H * ARG2H ->
                           ; PRODH:PRODL
   MOVFF   PRODH, RES3     ;
   MOVFF   PRODL, RES2     ;
;
   MOVF    ARG1L, W
   MULWF   ARG2H           ; ARG1L * ARG2H ->
                           ; PRODH:PRODL
   MOVF    PRODL, W        ;
   ADDWF   RES1, F         ; Add cross
   MOVF    PRODH, W        ; products
   ADDWFC  RES2, F         ;
   CLRF    WREG            ;
   ADDWFC  RES3, F         ;
;
   MOVF    ARG1H, W        ;
   MULWF   ARG2L           ; ARG1H * ARG2L ->
                           ; PRODH:PRODL
   MOVF    PRODL, W        ;
   ADDWF   RES1, F         ; Add cross
   MOVF    PRODH, W        ; products
   ADDWFC  RES2, F         ;
   CLRF    WREG            ;
   ADDWFC  RES3, F         ;
;
   BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
   BRA     SIGN_ARG1       ; no, check ARG1
   MOVF    ARG1L, W        ;
   SUBWF   RES2            ;
   MOVF    ARG1H, W        ;
   SUBWFB  RES3
;
SIGN_ARG1
   BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
   BRA     CONT_CODE       ; no, done
   MOVF    ARG2L, W        ;
   SUBWF   RES2            ;
   MOVF    ARG2H, W        ;
   SUBWFB  RES3
;
CONT_CODE
   :
```

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3:** External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Note: 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. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

9.6 INTx Pin Interrupts

External interrupts on the RB0/INT0, RB1/INT1, RB2/INT2 and RB3/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 RBx/INTx pin, the corresponding flag bit, INTxIF, is 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 power-managed modes if bit, INTxIE, was set prior to going into the power-managed modes. If the Global Interrupt Enable bit, GIE, is set, the processor will branch to the interrupt vector following wake-up.

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 11.0 "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 (ISR). 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 virtual bank
MOVFF   STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF   BSR, BSR_TEMP           ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF   BSR_TEMP, BSR           ; Restore BSR
MOVF    W_TEMP, W               ; Restore WREG
MOVFF   STATUS_TEMP, STATUS     ; Restore STATUS
```

TABLE 10-7: PORTC FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RC0/T1OSO/ T13CKI	RC0	0	O	DIG	LATC<0> data output.
		1	I	ST	PORTC<0> data input.
	T1OSO	x	O	ANA	Timer1 oscillator output; enabled when Timer1 oscillator enabled. Disables digital I/O and LCD segment driver.
	T13CKI	1	I	ST	Timer1/Timer3 counter input.
RC1/T1OSI/ CCP2/SEG32	RC1	0	O	DIG	LATC<1> data output.
		1	I	ST	PORTC<1> data input.
	T1OSI	x	I	ANA	Timer1 oscillator input.
	CCP2(1)	0	O	DIG	CCP2 compare/PWM output.
		1	I	ST	CCP2 capture input.
	SEG32	x	O	ANA	LCD Segment 32 output; disables all other pin functions.
RC2/CCP1/ SEG13	RC2	0	O	DIG	LATC<2> data output.
		1	I	ST	PORTC<2> data input.
	CCP1	0	O	DIG	CCP1 compare/PWM output; takes priority over port data.
		1	I	ST	CCP1 capture input.
	SEG13	x	O	ANA	LCD Segment 13 output; disables all other pin functions.
RC3/SCK/SCL/ SEG17	RC3	0	O	DIG	LATC<3> data output.
		1	I	ST	PORTC<3> data input.
	SCK	0	O	DIG	SPI clock output (MSSP module); takes priority over port data.
		1	I	ST	SPI clock input (MSSP module).
	SCL	0	O	DIG	I^2C™ clock output (MSSP module); takes priority over port data.
		1	I	I2C	I^2C clock input (MSSP module); input type depends on module setting.
	SEG17	x	O	ANA	LCD Segment 17 output; disables all other pin functions.
RC4/SDI/SDA/ SEG16	RC4	0	O	DIG	LATC<4> data output.
		1	I	ST	PORTC<4> data input.
	SDI		I	ST	SPI data input (MSSP module).
	SDA	1	O	DIG	I^2C data output (MSSP module); takes priority over port data.
		1	I	I2C	I^2C data input (MSSP module); input type depends on module setting.
	SEG16	x	O	ANA	LCD Segment 16 output; disables all other pin functions.
RC5/SDO/ SEG12	RC5	0	O	DIG	LATC<5> data output.
		1	I	ST	PORTC<5> data input.
	SDO	0	O	DIG	SPI data output (MSSP module).
	SEG12	x	O	ANA	LCD Segment 12 output; disables all other pin functions.
RC6/TX1/CK1/ SEG27	RC6	0	O	DIG	LATC<6> data output.
		1	I	ST	PORTC<6> data input.
	TX1	1	O	DIG	Synchronous serial data output (EUSART module); takes priority over port data.
	CK1	1	O	DIG	Synchronous serial data input (EUSART module); user must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSART module).
	SEG27	x	O	ANA	LCD Segment 27 output; disables all other pin functions.
RC7/RX1/DT1/ SEG28	RC7	0	O	DIG	LATC<7> data output.
		1	I	ST	PORTC<7> data input.
	RX1	1	I	ST	Asynchronous serial receive data input (EUSART module).
	DT1	1	O	DIG	Synchronous serial data output (EUSART module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSART module); user must configure as an input.
	SEG28	x	O	ANA	LCD Segment 28 output; disables all other pin functions.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, TTL = TTL Buffer Input, I2C = I^2C/SMBus Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Default assignment for CCP2 when CCP2MX Configuration bit is set.

15.1.1 RTCC CONTROL REGISTERS

REGISTER 15-1: RTCCFG: RTCC CONFIGURATION REGISTER(1)

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN(2)	—	RTCWREN	RTCSYNC	HALFSEC(3)	RTCOE	RTCPTR1	RTCPTR0
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 **RTCEN:** RTCC Enable bit(2)
1 = RTCC module is enabled
0 = RTCC module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **RTCWREN:** RTCC Value Registers Write Enable bit
1 = RTCVALH and RTCVALL registers can be written to by the user
0 = RTCVALH and RTCVALL registers are locked out from being written to by the user

bit 4 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
1 = RTCVALH, RTCVALL and ALRMRPT registers can change while reading due to a rollover ripple resulting in an invalid data read. If the register is read twice and results in the same data, the data can be assumed to be valid.
0 = RTCVALH, RTCVALL and ALCFGRPT registers can be read without concern over a rollover ripple

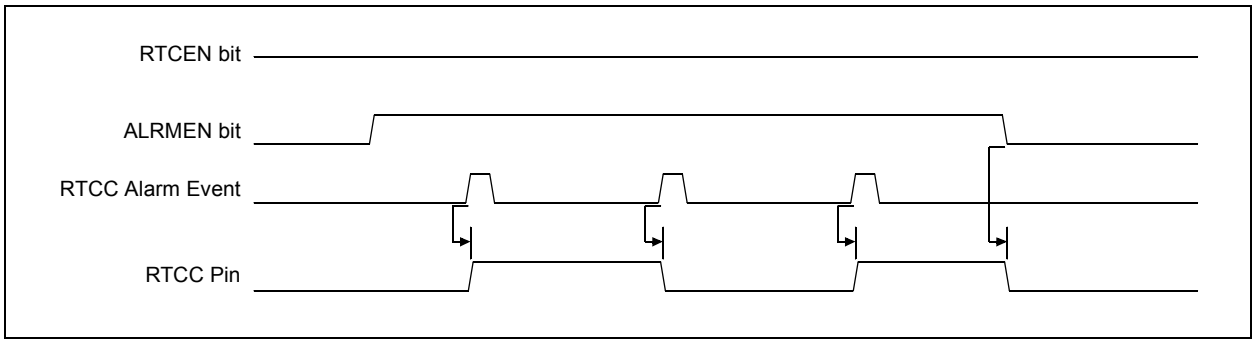
bit 3 **HALFSEC:** Half-Second Status bit(3)
1 = Second half period of a second
0 = First half period of a second

bit 2 **RTCOE:** RTCC Output Enable bit
1 = RTCC clock output enabled
0 = RTCC clock output disabled

bit 1-0 **RTCPTR<1:0>:** RTCC Value Register Window Pointer bits
Points to the corresponding RTCC Value registers when reading the RTCVALH and RTCVALL registers. The RTCPTR<1:0> value decrements on every read or write of RTCVALH<7:0> until it reaches '00'.
RTCVALH:
00 = Minutes
01 = Weekday
10 = Month
11 = Reserved
RTCVALL:
00 = Seconds
01 = Hours
10 = Day
11 = Year

Note 1: The RTCCFG register is only affected by a POR. For resets other than POR, RTCC will continue to run even if the device is in Reset.
2: A write to the RTCEN bit is only allowed when RTCWREN = 1.
3: This bit is read-only; it is cleared to '0' on a write to the lower half of the MINSEC register.

FIGURE 15-6: TIMER PULSE GENERATION



15.4 Sleep Mode

The timer and alarm continue to operate while in Sleep mode. The operation of the alarm is not affected by Sleep, as an alarm event can always wake-up the CPU.

The Idle mode does not affect the operation of the timer or alarm.

15.5 Reset

15.5.1 DEVICE RESET

When a device Reset occurs, the ALCFGRPT register is forced to its Reset state causing the alarm to be disabled (if enabled prior to the Reset). If the RTCC was enabled, it will continue to operate when a basic device Reset occurs.

15.5.2 POWER-ON RESET (POR)

The RTCCFG and ALRMRPT registers are reset only on a POR. Once the device exits the POR state, the clock registers should be reloaded with the desired values.

The timer prescaler values can be reset only by writing to the SECONDS register. No device Reset can affect the prescalers.

16.4.3 SETUP FOR PWM OPERATION

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

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR2L register and CCP2CON<5:4> bits.
3. Make the CCP2 pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
5. Configure the CCP2 module for PWM operation.

TABLE 16-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	59
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	60
PIR1	—	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	62
PIE1	—	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	62
IPR1	—	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	62
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	62
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	—	TRISE1	TRISE0	62
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	62
TMR2	Timer2 Register								60
PR2	Timer2 Period Register								60
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	60
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								63
CCPR1H	Capture/Compare/PWM Register 1 High Byte								63
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	63
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								64
CCPR2H	Capture/Compare/PWM Register 2 High Byte								63
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	64

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2.

17.2 LCD Clock Source

The LCD driver module generates its internal clock from 3 possible sources:

- System clock (FOSC/4)
- Timer1 oscillator
- INTRC source

The LCD clock generator uses a configurable divide-by-32/divide-by-8192 postscaler to produce a baseline frequency of about 1 kHz nominal, regardless of the source selected. The clock source selection and the postscaler configuration are determined by the Clock Source Select bits, CS<1:0> (LCDCON<3:2>).

An additional programmable prescaler is used to derive the LCD frame frequency from the 1 kHz baseline. The prescaler is configured using the LP<3:0> bits (LCDPS<3:0>) for any one of 16 options, ranging from 1:1 to 1:16.

Proper timing for waveform generation is set by the LMUX<1:0> bits (LCDCON<1:0>). These bits determine which Commons Multiplexing mode is to be used and divide down the LCD clock source as required. They also determine the configuration of the ring counter that is used to switch the LCD commons on or off.

17.2.1 LCD VOLTAGE REGULATOR CLOCK SOURCE

In addition to the clock source for LCD timing, a separate 31 kHz nominal clock is required for the LCD charge pump. This is provided from a distinct branch of the LCD clock source.

The charge pump clock can use either the Timer1 oscillator or the INTRC source, as well as the 8 MHz INTOSC source (after being divided by 256 by a prescaler). The charge pump clock source is configured using the CKSEL<1:0> bits (LCDREG<1:0>).

17.2.2 CLOCK SOURCE CONSIDERATIONS

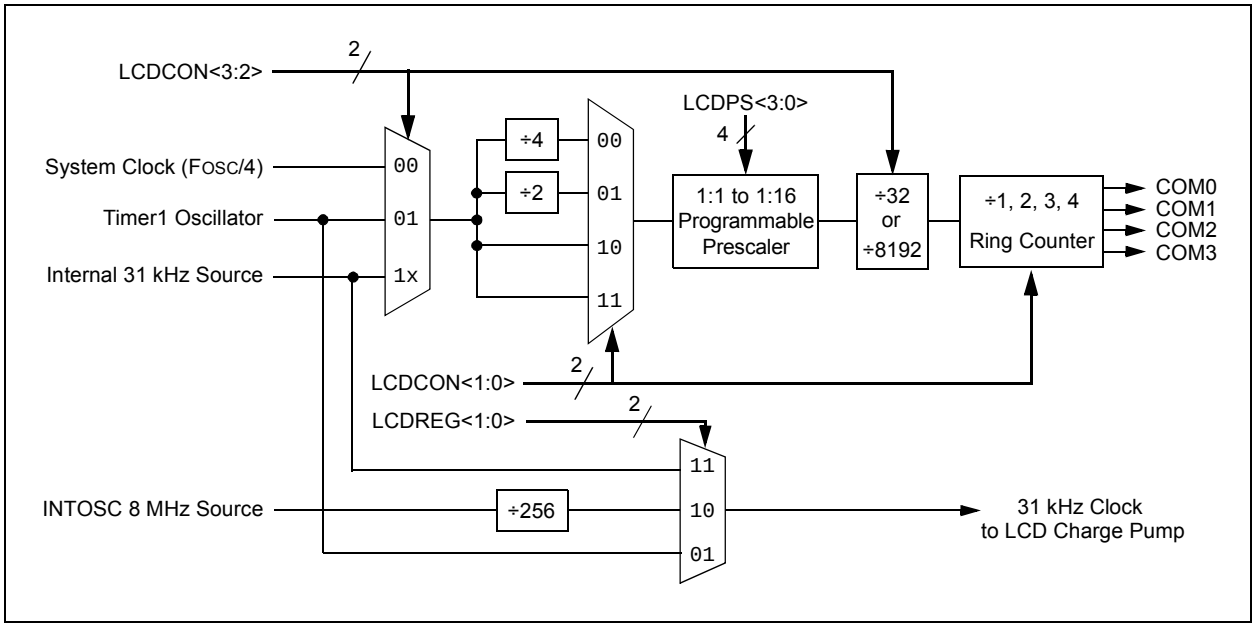
When using the system clock as the LCD clock source, it is assumed that the system clock frequency is a nominal 32 MHz (for a FOSC/4 frequency of 8 MHz). Because the prescaler option for the FOSC/4 clock selection is fixed at divide-by-8192, system clock speeds that differ from 32 MHz will produce frame frequencies and refresh rates different than discussed in this chapter. The user will need to keep this in mind when designing the display application.

The Timer1 and INTRC sources can be used as LCD clock sources when the device is in Sleep mode. To use the Timer1 oscillator, it is necessary to set the T1OSCEN bit (T1CON<3>). Selecting either Timer1 or INTRC as the LCD clock source will not automatically activate these sources.

Similarly, selecting the INTOSC as the charge pump clock source will not turn the oscillator on. To use INTOSC, it must be selected as the system clock source by using the FOSC2 Configuration bit.

If Timer1 is used as a clock source for the device, either as an LCD clock source or for any other purpose, LCD segment 32 become unavailable.

FIGURE 17-2: LCD CLOCK GENERATION



18.3.3 ENABLING SPI I/O

To enable the serial port, the MSSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$ must have TRISF<7> bit set

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

18.3.4 OPEN-DRAIN OUTPUT OPTION

The drivers for the SDO output and SCK clock pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters.

The open-drain output option is controlled by the SPIOD bit (TRISG<7>). Setting this bit configures both pins for open-drain operation.

18.3.5 TYPICAL CONNECTION

Figure 18-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

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

FIGURE 18-2: SPI MASTER/SLAVE CONNECTION

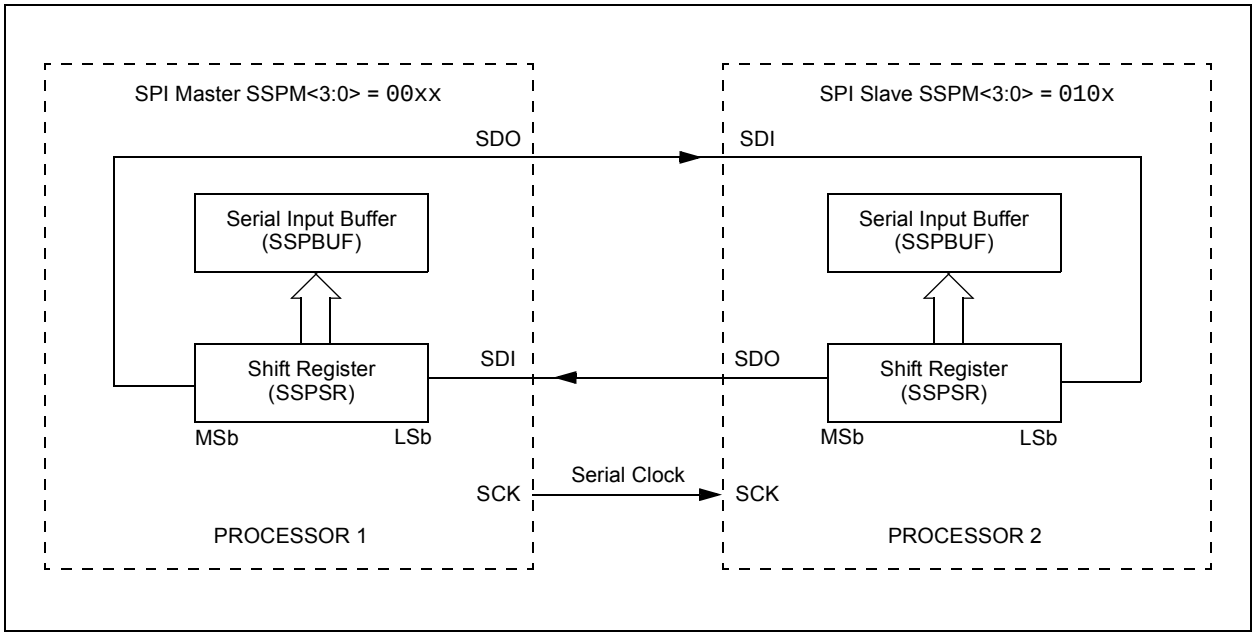
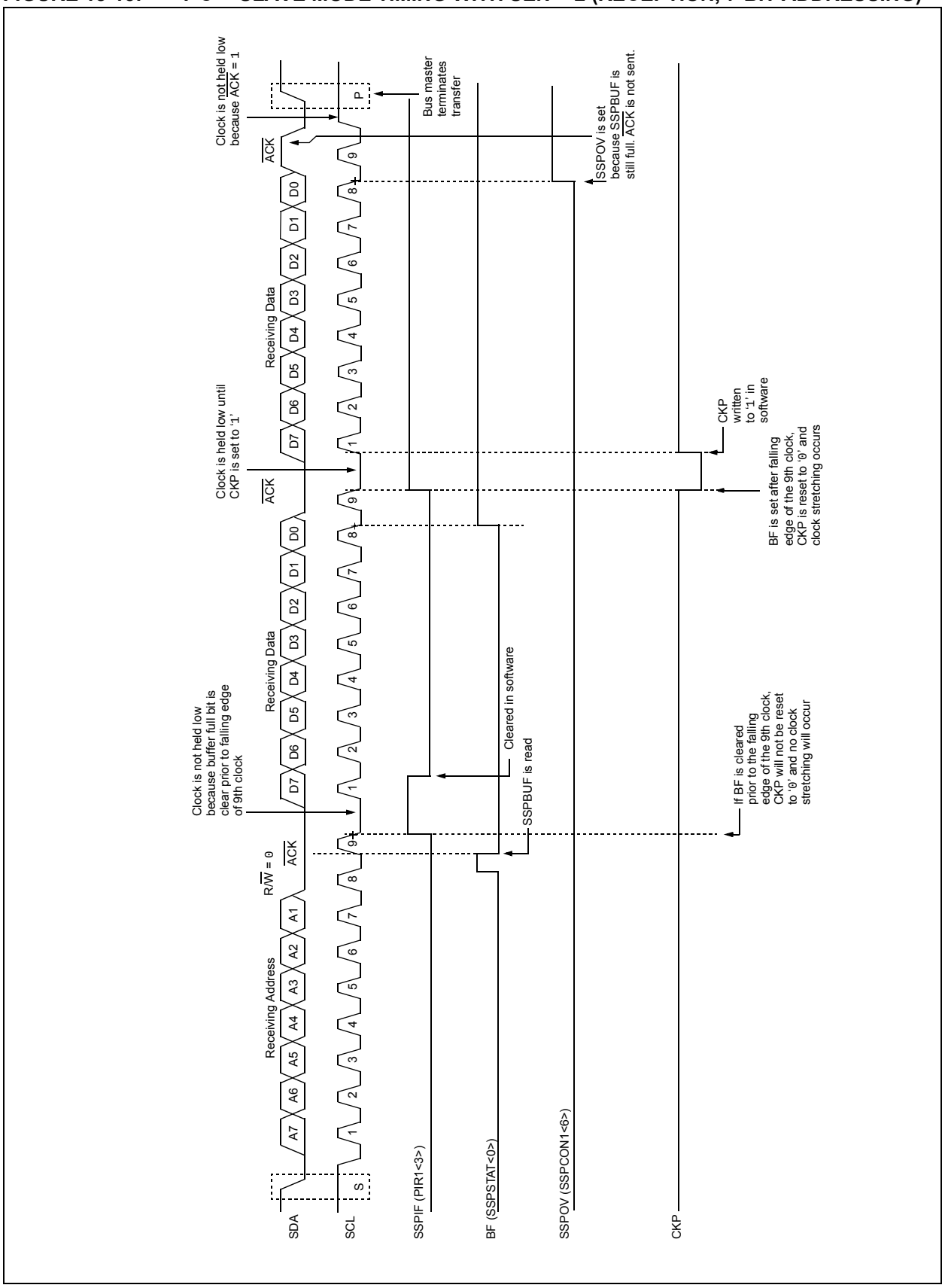


FIGURE 18-15: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESSING)



18.4.8 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA pin being driven low while SCL is high is the Start condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware. The Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

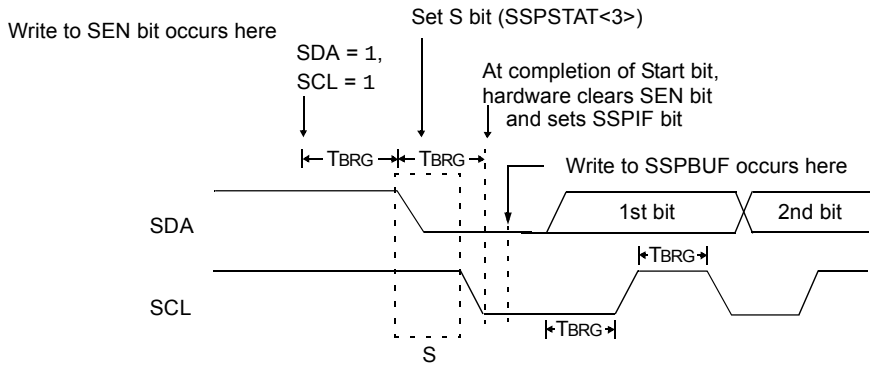
Note:	If, at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and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

18.4.8.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete.

FIGURE 18-21: FIRST START BIT TIMING



24.3.2 CAPACITANCE CALIBRATION

There is a small amount of capacitance from the internal A/D Converter sample capacitor as well as stray capacitance from the circuit board traces and pads that affect the precision of capacitance measurements. A measurement of the stray capacitance can be taken by making sure the desired capacitance to be measured has been removed. The measurement is then performed using the following steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT (= 1).
3. Wait for a fixed delay of time, t.
4. Clear EDG1STAT.
5. Perform an A/D conversion.
6. Calculate the stray and A/D sample capacitances:

$$C_{OFFSET} = C_{STRAY} + C_{AD} = (I \bullet t)/V$$

where I is known from the current source measurement step, t is a fixed delay and V is measured by performing an A/D conversion.

This measured value is then stored and used for calculations of time measurement or subtracted for capacitance measurement. For calibration, it is expected that the capacitance of $C_{STRAY} + C_{AD}$ is approximately known; C_{AD} is approximately 4 pF.

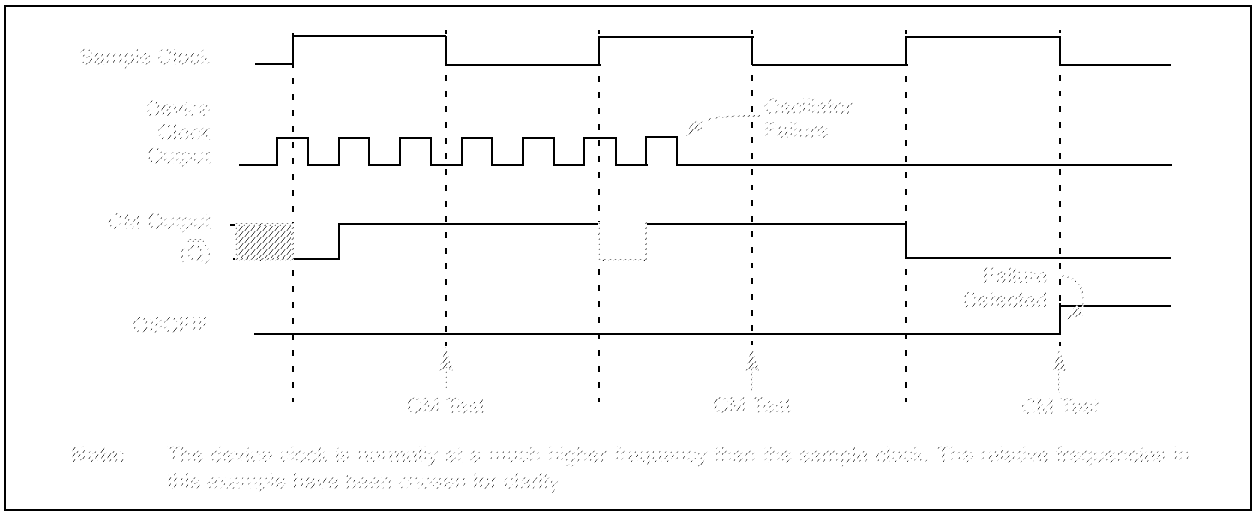
An iterative process may need to be used to adjust the time, t, that the circuit is charged to obtain a reasonable voltage reading from the A/D Converter. The value of t may be determined by setting C_{OFFSET} to a theoretical value, then solving for t. For example, if C_{STRAY} is theoretically calculated to be 11 pF, and V is expected to be 70% of V_{DD}, or 2.31V, then t would be:

$$(4\ pF + 11\ pF) \bullet 2.31V/0.55\ \mu A$$

or 63 μs.

See Example 24-3 for a typical routine for CTMU capacitance calibration.

FIGURE 25-5: FSCM TIMING DIAGRAM



25.5.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 2H (with any required start-up delays that are required for the oscillator mode, such as the OST or PLL timer). The INTRC oscillator provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexor. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

25.5.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexor selects the clock source selected by the OSCCON register. Fail-Safe Clock Monitoring of the power-managed clock source resumes in the power-managed mode.

If an oscillator failure occurs during power-managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTRC multiplexor. An automatic transition back to the failed clock source will not occur.

25.5.4 POR OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is either EC or INTRC mode, monitoring can begin immediately following these events.

For HS or HSPLL modes, the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

Note:	The same logic that prevents false oscillator failure interrupts on POR, or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 25.4.1 "Special Considerations for Using Two-Speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary clock to become stable. When the new power-managed mode is selected, the primary clock is disabled.

26.2.2 EXTENDED INSTRUCTION SET

ADDFSR **Add Literal to FSR**

Syntax: ADDFSR f, k

Operands: $0 \le k \le 63$
$f \in [0, 1, 2]$

Operation: FSR(f) + k → FSR(f)

Status Affected: None

Encoding:

1110	1000	ffkk	kkkk

Description: The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR

Example: ADDFSR 2, 23h

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

ADDULNK **Add Literal to FSR2 and Return**

Syntax: ADDULNK k

Operands: $0 \le k \le 63$

Operation: FSR2 + k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:

1110	1000	11kk	kkkk

Description: The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a NOP is performed during the second cycle.

This may be thought of as a special case of the ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

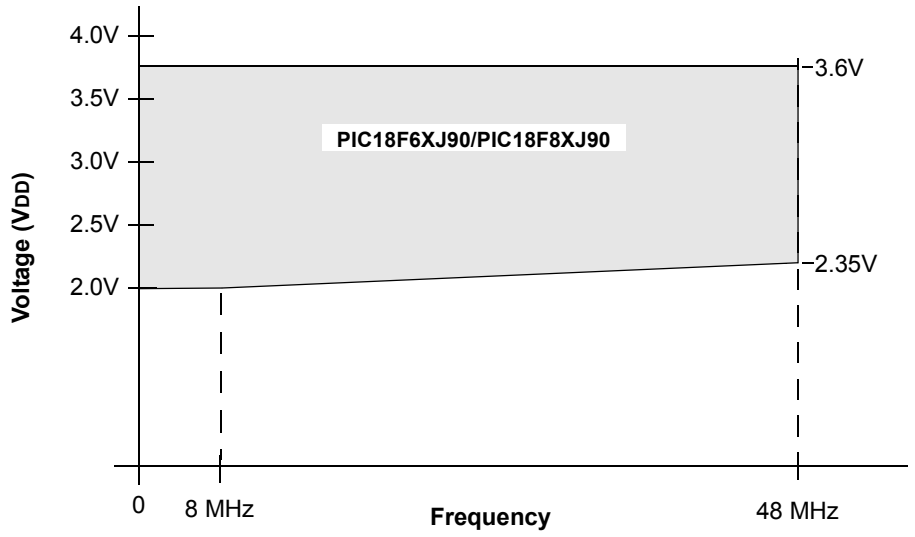
Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: ADDULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 0422h
PC = (TOS)

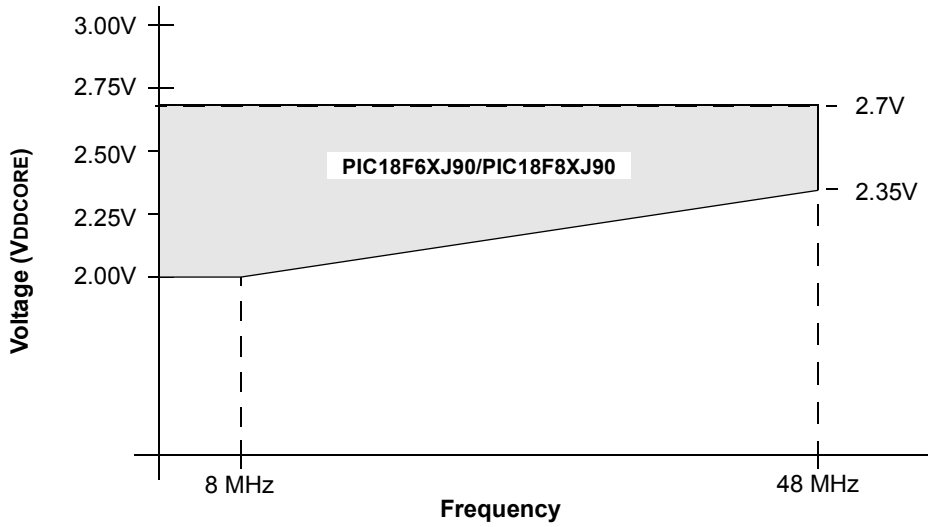
Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

FIGURE 28-1: VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL)(1)



Note 1: When the on-chip regulator is enabled, its BOR circuit will automatically trigger a device Reset before VDD reaches a level at which full-speed operation is not possible.

FIGURE 28-2: VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL)(1,2)



Note 1: When the on-chip voltage regulator is disabled, VDD and VDDCORE must be maintained so that VDDCORE ≤ VDD ≤ 3.6V.

28.2 DC Characteristics: Power-Down and Supply Current PIC18F87J90 Family (Industrial) (Continued)

PIC18F87J90 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD)(2,3)						
	All devices	5	14.2	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 31 kHz (**RC_RUN** mode, internal oscillator source)
		5.5	14.2	µA	+25°C		
		10	19.0	µA	+85°C		
	All devices	6.8	16.5	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		7.6	16.5	µA	+25°C		
		14	22.4	µA	+85°C		
	All devices	37	84	µA	-40°C	VDD = 3.3V(5)	
		51	84	µA	+25°C		
		72	108	µA	+85°C		
	All devices	0.43	0.82	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (**RC_RUN** mode, internal oscillator source)
		0.47	0.82	mA	+25°C		
		0.52	0.95	mA	+85°C		
	All devices	0.52	0.98	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.57	0.98	mA	+25°C		
		0.63	1.10	mA	+85°C		
	All devices	0.59	0.96	mA	-40°C	VDD = 3.3V(5)	
		0.65	0.96	mA	+25°C		
		0.72	1.18	mA	+85°C		
	All devices	0.88	1.45	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (**RC_RUN** mode, internal oscillator source)
		1	1.45	mA	+25°C		
		1.1	1.58	mA	+85°C		
	All devices	1.2	1.72	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		1.3	1.72	mA	+25°C		
		1.4	1.85	mA	+85°C		
	All devices	1.3	2.87	mA	-40°C	VDD = 3.3V(5)	
		1.4	2.87	mA	+25°C		
		1.5	2.96	mA	+85°C		

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

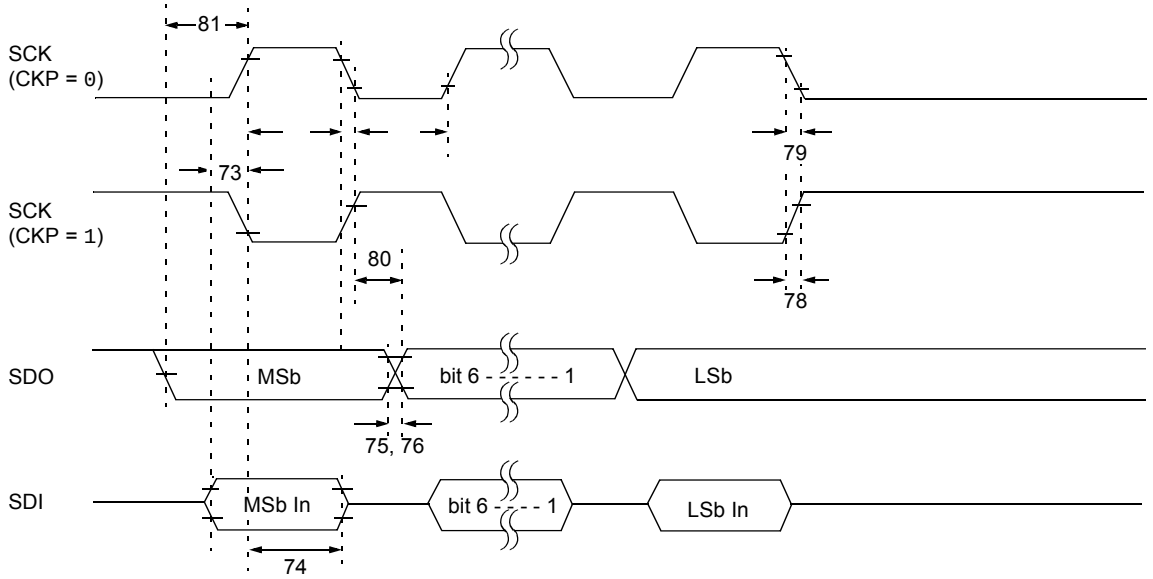
2: The supply current is mainly a function of the operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all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

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

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

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

FIGURE 28-10: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)



Note: Refer to Figure 28-3 for load conditions.

TABLE 28-15: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TDIV2SCH, TDIV2SCL	Setup Time of SDI Data Input to SCK Edge	20	—	ns	
73A	TB2B	Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2	1.5 TCY + 40	—	ns	**(Note 2)**
74	TSCH2DIL, TSCL2DIL	Hold Time of SDI Data Input to SCK Edge	40	—	ns	
75	TDOR	SDO Data Output Rise Time	—	25	ns	
76	TDOF	SDO Data Output Fall Time	—	25	ns	
78	TSCR	SCK Output Rise Time (Master mode)	—	25	ns	
79	TSCF	SCK Output Fall Time (Master mode)	—	25	ns	
80	TSCH2DOV, TSCL2DOV	SDO Data Output Valid after SCK Edge	—	50	ns	
81	TDOV2SCH, TDOV2SCL	SDO Data Output Setup to SCK Edge	TCY	—	ns	

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.