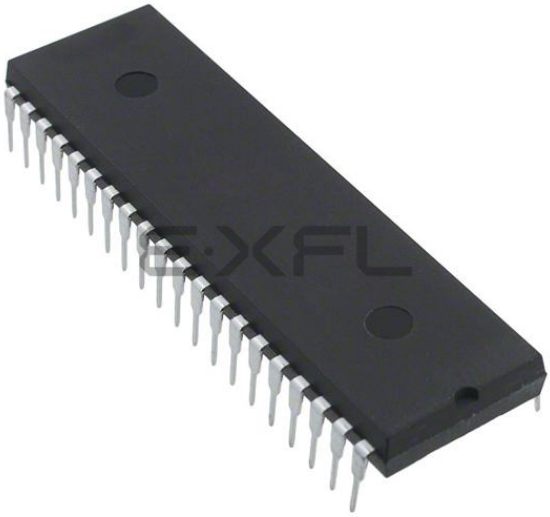


Welcome to E-XFL.COM

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	64MHz
Connectivity	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f45k80-i-p

NOTES:

TABLE 1-4: PIC18F2XK80 I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	QFN	SSOP/ SPDIP /SOIC			
$\overline{\text{MCLR}}$/RE3	26	1			
MCLR			I	ST	Master Clear (input) or programming voltage (input). This pin is an active-low Reset to the device.
RE3			I	ST	General purpose, input only pin.
OSC1/CLKIN/RA7	6	9			
OSC1			I	ST	Oscillator crystal input.
CLKIN			I	CMOS	External clock source input. Always associated with pin function, OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
RA7			I/O	ST/ CMOS	General purpose I/O pin.
OSC2/CLKOUT/RA6	7	10			
OSC2			O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKOUT			O	—	In certain oscillator modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6			I/O	ST/ CMOS	General purpose I/O pin.

Legend: CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
I^2C™ = I^2C/SMBus input buffer
Analog = Analog input
O = Output

TABLE 1-6: PIC18F6XK80 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Num	Pin Type	Buffer Type	Description
				PORTD is a bidirectional I/O port.
RD0/C1INA/PSP0	54			
RD0		I/O	ST/ CMOS	Digital I/O.
C1INA		I	Analog	Comparator 1 Input A.
PSP0		I/O	ST/ CMOS	Parallel Slave Port data.
RD1/C1INB/PSP1	55			
RD1		I/O	ST/ CMOS	Digital I/O.
C1INB		I	Analog	Comparator 1 Input B.
PSP1		I/O	ST/ CMOS	Parallel Slave Port data.
RD2/C2INA/PSP2	58			
RD2		I/O	ST/ CMOS	Digital I/O.
C2INA		I	Analog	Comparator 2 Input A.
PSP2		I/O	ST/ CMOS	Parallel Slave Port data.
RD3/C2INB/CTMUI/ PSP3	59			
RD3		I/O	ST/ CMOS	Digital I/O.
C2INB		I	Analog	Comparator 2 Input B.
CTMUI		O	CMOS	CTMU pulse generator charger for the C2INB.
PSP3		I/O	ST/ CMOS	Parallel Slave Port data.
RD4/ECCP1/P1A/PSP4	2			
RD4		I/O	ST/ CMOS	Digital I/O.
ECCP1		I/O	ST	Capture 1 input/Compare 1 output/PWM1 output.
P1A		O	CMOS	Enhanced PWM1 Output A.
PSP4		I/O	ST/ CMOS	Parallel Slave Port data.
RD5/P1B/PSP5	3			
RD5		I/O	ST/ CMOS	Digital I/O.
P1B		O	CMOS	Enhanced PWM1 Output B.
PSP5		I/O	ST/ CMOS	Parallel Slave Port data.

Legend: I^2C™ = I^2C/SMBus input buffer
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output

5.6.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (Parameter 33). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HSPLL modes and only on Power-on Reset or on exit from most power-managed modes.

5.6.3 PLL LOCK TIME-OUT

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (T_{PLL}) is typically 2 ms and follows the oscillator start-up time-out.

5.6.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

1. After the POR pulse has cleared, PWRT time-out is invoked (if enabled).
2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 5-3, Figure 5-4, Figure 5-5, Figure 5-6 and Figure 5-7 all depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figures 5-3 through 5-6 also apply to devices operating in XT or LP modes. For devices in RC mode and with the PWRT disabled, on the other hand, there will be no time-out at all.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, all time-outs will expire. Bringing $\overline{MCLR}$ high will begin execution immediately (Figure 5-5). This is useful for testing purposes or to synchronize more than one PIC18FXXXX device operating in parallel.

TABLE 5-2: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up and Brown-out		Exit from Power-Managed Mode
	$\overline{PWRTEN}$ = 0	$\overline{PWRTEN}$ = 1	
HSPLL	66 ms(1) + 1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)
HS, XT, LP	66 ms(1) + 1024 TOSC	1024 TOSC	1024 TOSC
EC, ECIO	66 ms(1)	—	—
RC, RCIO	66 ms(1)	—	—
INTIO1, INTIO2	66 ms(1)	—	—

Note 1: 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.

2: 2 ms is the nominal time required for the PLL to lock.

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD}, V_{DD} RISE < T_{PWRT})

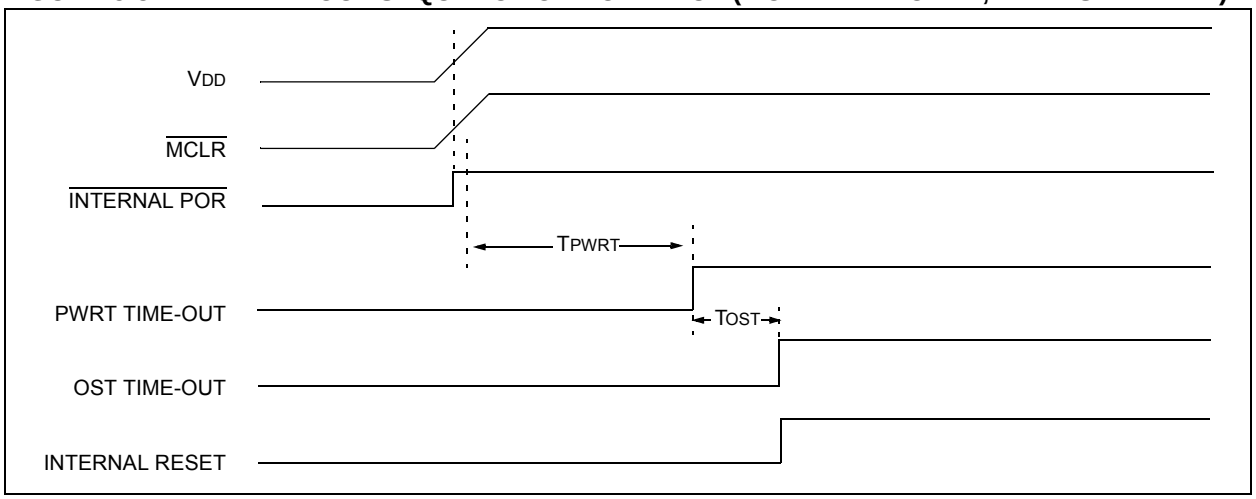


TABLE 5-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
CCPTMRS	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	---0 0000	---x xxxx	---u uuuu
TRISG	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	---1 1111	---1 1111	---u uuuu
TRISF	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
TRISE	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 -111	1111 -111	uuuu -uuu
TRISD	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
TRISA(5)	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	111- 1111(5)	111- 1111(5)	uuu- uuuu(5)
ODCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
SLRCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-000 0000	-111 1111	-111 1111
LATG	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	---x xxxx	---x xxxx	---u uuuu
LATF	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx -xxx	uuuu -uuu
LATE	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx -xxx	xxxx xxxx	uuuu uuuu
LATD	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
LATC	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
LATB	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
LATA(5)	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxx- xxxx(5)	xxx- xxxx(5)	uuu- uuuu(5)
T4CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-000 0000	-000 0000	-uuu uuuu
TMR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
PORTG	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	---x xxxx	---x xxxx	---u uuuu
PORTF	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTE	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTD	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTC	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTB	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	xxxx xxxx	uuuu uuuu
PORTA(5)	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxx- xxxx(5)	xxx- xxxx(5)	uuu- uuuu(5)
EECON1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xx-0 x000	uu-0 u000	uu-u uuuu
EECON2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
SPBRGH1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
SPBRGH2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
SPBRG2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
IPR5	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
PIR5	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged; x = unknown; - = unimplemented bit, read as '0'; q = value depends on condition.
Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

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: 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.

4: See Table 5-3 for Reset value for specific conditions.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

TABLE 6-2: PIC18F66K80 FAMILY REGISTER FILE SUMMARY (CONTINUED)

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR on page
F9Fh	IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	TMR1GIP	TMR2IP	TMR1IP	90
F9Eh	PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	TMR1GIF	TMR2IF	TMR1IF	89
F9Dh	PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	TMR1GIE	TMR2IE	TMR1IE	89
F9Ch	PSTR1CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA	89
F9Bh	OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	90
F9Ah	REFOCON	ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	90
F99h	CCPTMRS	—	—	—	C5TSEL	C4TSEL	C3TSEL	C2TSEL	C1TSEL	90
F98h	TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	90
F97h	TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	90
F96h	TRISE	TRISE7	TRISE6	TRISE5	TRISE4	—	TRISE2	TRISE1	TRISE0	90
F95h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	90
F94h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	90
F93h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	90
F92h	TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	90
F91h	ODCON	SSPOD	CCP5OD	CCP4OD	CCP3OD	CCP2OD	CCP1OD	U2OD	U1OD	90
F90h	SLRCON	—	SLRG	SLRF	SLRE	SLRD	SLRC	SLRB	SLRA	90
F8Fh	LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	90
F8Eh	LATF	LATF7	LATF6	LATF5	LATF4	—	LATF2	LATF1	LATF0	90
F8Dh	LATE	LATE7	LATE6	LATE5	LATE4	—	LATE2	LATE1	LATE0	90
F8Ch	LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	90
F8Bh	LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	90
F8Ah	LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	90
F89h	LATA	LATA7	LATA6	LATA5	—	LATA3	LATA2	LATA1	LATA0	90
F88h	T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	90
F87h	TMR4	Timer4 Register								90
F86h	PORTG	—	—	—	RG4	RG3	RG2	RG1	RG0	90
F85h	PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	90
F84h	PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	90
F83h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	90
F82h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	90
F81h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	90
F80h	PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	90
F7Fh	EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	90
F7Eh	EECON2	Flash Self-Program Control Register (not a physical register)								90
F7Dh	SPBRGH1	EUSART1 Baud Rate Generator Register High Byte								90
F7Ch	SPBRGH2	EUSART2 Baud Rate Generator Register High Byte								90
F7Bh	SPBRG2	EUSART2 Baud Rate Generator Register Low Byte								90
F7Ah	RCREG2	EUSART2 Receive Register								90
F79h	TXREG2	EUSART2 Transmit Register								91
F78h	IPR5	IRXIP	WAKIP	ERRIP	TX2BIP	TXB1IP	TXB0IP	RXB1IP	RXB0IP	91
F77h	PIR5	IRXIF	WAKIF	ERRIF	TXB2IF	TXB1IF	TXB0IF	RXB1IF	RXB0IF	91
F76H	PIE5	IRXIE	WAKIE	ERRIE	TX2BIE	TXB1IE	TXB0IE	RXB1IE	RXB0IE	91
F75h	EEADRH	Data EE Address Register High Byte								91
F74h	EEADR	Data EE Address Register Low Byte								91
F73h	EEDATA	Data EE Data Register								91
F72h	ECANCON	MDSEL1	MDSEL0	FIFOWM	EWIN4	EWIN3	EWIN2	EWIN1	EWIN0	91
F71h	COMSTAT	RXB0OVFL	RXB1OVFL	TXBO	TXBP	RXBP	TXWARN	RXWARN	EWARN	91
F70h	CIOCON	TX2SRC	TX2EN	ENDRHI	CANCAP	—	—	—	CLKSEL	91
F6Fh	CANCON	REQOP2	REQOP1	REQOP0	ABAT	WIN2/FP3	WIN1/FP2	WIN0/FP1	FP0	91
F6Eh	CANSTAT	OPMODE2	OPMODE1	OPMODE0	—/ EICOD4	ICODE2/ EICODE3	ICODE1/ EICODE2	ICODE0/ EICODE1	—/ EICODE0	91

EXAMPLE 8-1: DATA EEPROM READ

```
        MOVLW   DATA_EE_ADDRH      ;
        MOVWF   EEADRH             ; Upper bits of Data Memory Address to read
        MOVLW   DATA_EE_ADDR       ;
        MOVWF   EEADR              ; Lower bits of Data Memory Address to read
        BCF     EECON1, EEPGD      ; Point to DATA memory
        BCF     EECON1, CFGS       ; Access EEPROM
        BSF     EECON1, RD         ; EEPROM Read
        NOP
        MOVF    EEDATA, W          ; W = EEDATA
```

EXAMPLE 8-2: DATA EEPROM WRITE

```
                MOVLW   DATA_EE_ADDRH      ;
                MOVWF   EEADRH             ; Upper bits of Data Memory Address to write
                MOVLW   DATA_EE_ADDR       ;
                MOVWF   EEADR              ; Lower bits of Data Memory Address to write
                MOVLW   DATA_EE_DATA       ;
                MOVWF   EEDATA             ; Data Memory Value to write
                BCF     EECON1, EEPGD      ; Point to DATA memory
                BCF     EECON1, CFGS       ; Access EEPROM
                BSF     EECON1, WREN       ; Enable writes

                BCF     INTCON, GIE        ; Disable Interrupts
                MOVLW   55h                ;
Required        MOVWF   EECON2             ; Write 55h
Sequence        MOVLW   0AAh               ;
                MOVWF   EECON2             ; Write 0AAh
                BSF     EECON1, WR         ; Set WR bit to begin write
                BTFSC   EECON1, WR         ; Wait for write to complete GOTO $-2
                BSF     INTCON, GIE        ; Enable Interrupts

                                           ; User code execution
                BCF     EECON1, WREN       ; Disable writes on write complete (EEIF set)
```

8.6 Operation During Code-Protect

Data EEPROM memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if code protection is enabled.

The microcontroller itself can both read and write to the internal data EEPROM regardless of the state of the code-protect Configuration bit. Refer to **Section 28.0 "Special Features of the CPU"** for additional information.

8.7 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been implemented. On power-up, the WREN bit is cleared. In addition, writes to the EEPROM are blocked during the Power-up Timer period (T_{PWRT}, Parameter 33).

The write initiate sequence, and the WREN bit together, help prevent an accidental write during brown-out, power glitch or software malfunction.

8.8 Using the Data EEPROM

The data EEPROM is a high-endurance, byte-addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). Frequently changing values will typically be updated more often than Parameter D124. If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 8-3.

Note:	If data EEPROM is only used to store constants and/or data that changes often, an array refresh is likely not required. See Parameter D124.

EXAMPLE 8-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADR             ; Start at address 0
        CLRF    EEADRH            ;
        BCF     EECON1, CFGS      ; Set for memory
        BCF     EECON1, EEPGD     ; Set for Data EEPROM
        BCF     INTCON, GIE       ; Disable interrupts
        BSF     EECON1, WREN      ; Enable writes
LOOP                              ; Loop to refresh array
        BSF     EECON1, RD        ; Read current address
        MOVLW   55h               ;
        MOVWF   EECON2            ; Write 55h
        MOVLW   0AAh              ;
        MOVWF   EECON2            ; Write 0AAh
        BSF     EECON1, WR        ; Set WR bit to begin write
        BTFSC   EECON1, WR        ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F          ; Increment address
        BRA     LOOP              ; Not zero, do it again
        INCFSZ  EEADRH, F         ; Increment the high address
        BRA     LOOP              ; Not zero, do it again

        BCF     EECON1, WREN      ; Disable writes
        BSF     INTCON, GIE       ; Enable interrupts
```

TABLE 11-9: PORTE FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE5/CANTX	RE5(1)	0	O	DIG	LATE<5> data output.
		1	I	ST	PORTE<5> data input.
	CANTX(1,2)	0	O	DIG	CAN bus TX.
RE6/RX2/DT2	RE6(1)	0	O	DIG	LATE<6> data output.
		1	I	ST	PORTE<6> data input.
	RX2(1)	1	I	ST	Asynchronous serial receive data input (EUSARTx module).
	DT2(1)	1	O	DIG	Synchronous serial data output (EUSARTx module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSARTx module); user must configure as an input.
RE7/TX2/CK2	RE7(1)	0	O	DIG	LATE<7> data output.
		1	I	ST	PORTE<7> data input.
	TX2(1)	0	O	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.
	CK2(1)	0	O	DIG	Synchronous serial clock output (EUSARTx module); user must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSARTx module); user must configure as an input.

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

Note 1: These bits are unavailable for 40 and 44-pin devices (PIC18F4XK0).

2: This is the alternate pin assignment for CANRX and CANTX on 64-pin devices (PIC18F6XK80) when the CANMX Configuration bit is cleared.

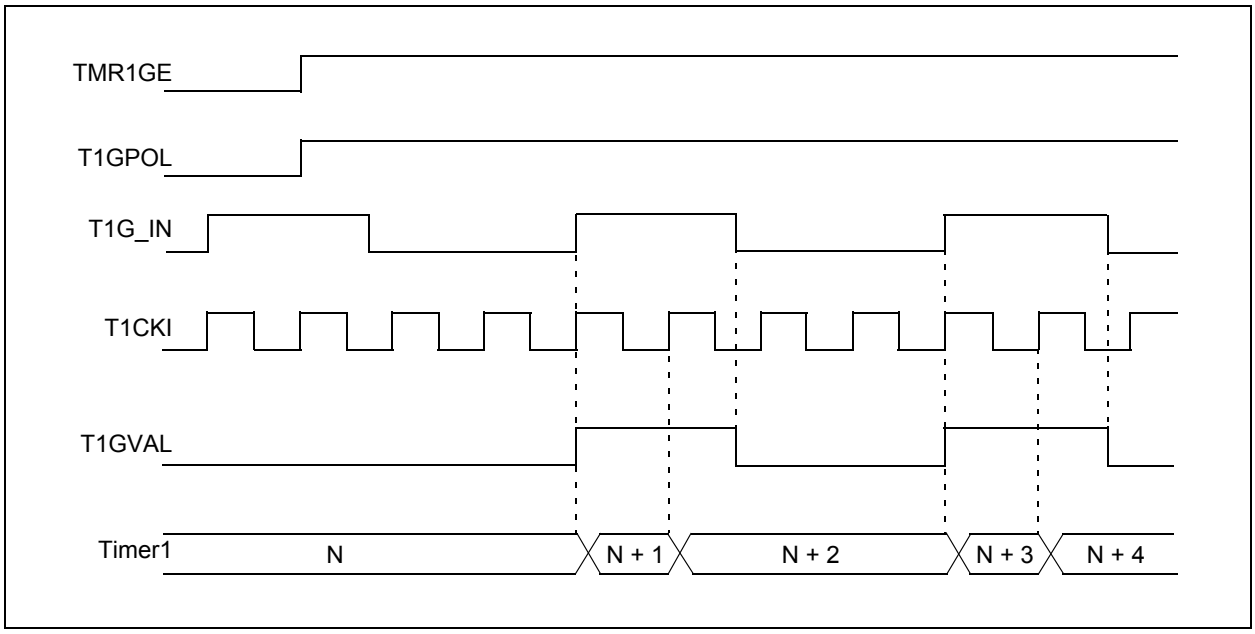
TABLE 11-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTE	RE7(1)	RE6(1)	RE5(1)	RE4(1)	RE3	RE2	RE1	RE0
LATE	LATE7	LATE6	LATE5	LATE4	—	LATE2	LATE1	LATE0
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	—	TRISE2	TRISE1	TRISE0
PADCFG1	RDPU	REPU	RFPU(1)	RGPU(1)	—	—	—	CTMUDS
ANCON0	ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0

Legend: Shaded cells are not used by PORTE.

Note 1: These bits are unimplemented on 44-pin devices, read as '0'.

FIGURE 14-4: TIMER1 GATE COUNT ENABLE MODE



14.8.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four sources. Source selection is controlled by the T1GSSx (T1GCON<1:0>) bits (see Table 14-4).

TABLE 14-4: TIMER1 GATE SOURCES

T1GSS<1:0>	Timer1 Gate Source
00	Timer1 Gate Pin
01	TMR2 to Match PR2 (TMR2 increments to match PR2)
10	Comparator 1 Output (comparator logic high output)
11	Comparator 2 Output (comparator logic high output)

The polarity for each available source is also selectable, controlled by the T1GPOL bit (T1GCON<6>).

14.8.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

14.8.2.2 Timer2 Match Gate Operation

The TMR2 register will increment until it matches the value in the PR2 register. On the very next increment cycle, TMR2 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry. The pulse will remain high for one instruction cycle and will return back to a low state until the next match.

Depending on T1GPOL, Timer1 increments differently when TMR2 matches PR2. When T1GPOL = 1, Timer1 increments for a single instruction cycle following a TMR2 match with PR2. When T1GPOL = 0, Timer1 increments continuously except for the cycle following the match when the gate signal goes from low-to-high.

14.8.2.3 Comparator 1 Output Gate Operation

The output of Comparator 1 can be internally supplied to the Timer1 gate circuitry. After setting up Comparator 1 with the CM1CON register, Timer1 will increment depending on the transitions of the CMP1OUT (CMSTAT<6>) bit.

14.8.2.4 Comparator 2 Output Gate Operation

The output of Comparator 2 can be internally supplied to the Timer1 gate circuitry. After setting up Comparator 2 with the CM2CON register, Timer1 will increment depending on the transitions of the CMP2OUT (CMSTAT<7>) bit.

REGISTER 21-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C™ MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
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 **WCOL:** Write Collision Detect bit

In Master Transmit mode:
1 = A write to the SSPBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)
0 = No collision

In Slave Transmit mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

In Receive mode (Master or Slave modes):
This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)
0 = No overflow

In Transmit mode:
This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(1)
1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCK Release Control bit

In Slave mode:
1 = Releases clock
0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:
Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(2)
1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1011 = I²C Firmware Controlled Master mode (slave Idle)
1001 = Load SSPMSK register at SSPADD SFR address(3,4)
1000 = I²C Master mode, clock = FOSC/(4 * (SSPADD + 1))
0111 = I²C Slave mode, 10-bit address
0110 = I²C Slave mode, 7-bit address

Note 1: When enabled, the SDA and SCL pins must be configured as inputs.
2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.
3: When SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address actually access the SSPMSK register.
4: This mode is only available when 7-Bit Address Masking mode is selected (MSSPMSK Configuration bit is '1').

21.4.3.4 7-Bit Address Masking Mode

Unlike 5-bit masking, 7-Bit Address Masking mode uses a mask of up to 8 bits (in 10-bit addressing) to define a range of addresses that can be Acknowledged, using the lowest bits of the incoming address. This allows the module to Acknowledge up to 127 different addresses with 7-bit addressing, or 255 with 10-bit addressing (see Example 21-3). This mode is the default configuration of the module, which is selected when MSSPMSK is unprogrammed ('1').

The address mask for 7-Bit Address Masking mode is stored in the SSPMSK register, instead of the SSPCON2 register. SSPMSK is a separate hardware register within the module, but it is not directly addressable. Instead, it shares an address in the SFR space with the SSPADD register. To access the SSPMSK register, it is necessary to select MSSP mode, '1001' (SSPCON1<3:0> = 1001) and then read or write to the location of SSPADD.

To use 7-Bit Address Masking mode, it is necessary to initialize SSPMSK with a value before selecting the I^2C Slave Addressing mode. Thus, the required sequence of events is:

1. Select SSPMSK Access mode (SSPCON2<3:0> = 1001).
2. Write the mask value to the appropriate SSPADD register address (FC8h).
3. Set the appropriate I^2C Slave mode (SSPCON2<3:0> = 0111 for 10-bit addressing, 0110 for 7-bit addressing).

Setting or clearing mask bits in SSPMSK behaves in the opposite manner of the ADMSKx bits in 5-Bit Address Masking mode. That is, clearing a bit in SSPMSK causes the corresponding address bit to be masked; setting the bit requires a match in that position. SSPMSK resets to all '1's upon any Reset condition and, therefore, has no effect on the standard MSSP operation until written with a mask value.

With 7-bit addressing, SSPMSK<7:1> bits mask the corresponding address bits in the SSPADD register. For any SSPMSK bits that are active (SSPMSK<n> = 0), the corresponding SSPADD address bit is ignored (SSPADD<n> = x). For the module to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

With 10-bit addressing, SSPMSK<7:0> bits mask the corresponding address bits in the SSPADD register. For any SSPMSK bits that are active (= 0), the corresponding SSPADD address bit is ignored (SSPADD<n> = x).

Note:	The two Most Significant bits of the address are not affected by address masking.

EXAMPLE 21-3: ADDRESS MASKING EXAMPLES IN 7-BIT MASKING MODE

7-Bit Addressing:

SSPADD<7:1> = 1010 000

SSPMSK<7:1> = 1111 001

Addresses Acknowledged = ACh, A8h, A4h, A0h

10-Bit Addressing:

SSPADD<7:0> = 1010 0000 (The two MSb are ignored in this example since they are not affected)

SSPMSK<5:1> = 1111 0011

Addresses Acknowledged = ACh, A8h, A4h, A0h

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion can start. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 23.3 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/$\overline{\text{DONE}}$ bit and the actual start of the conversion.

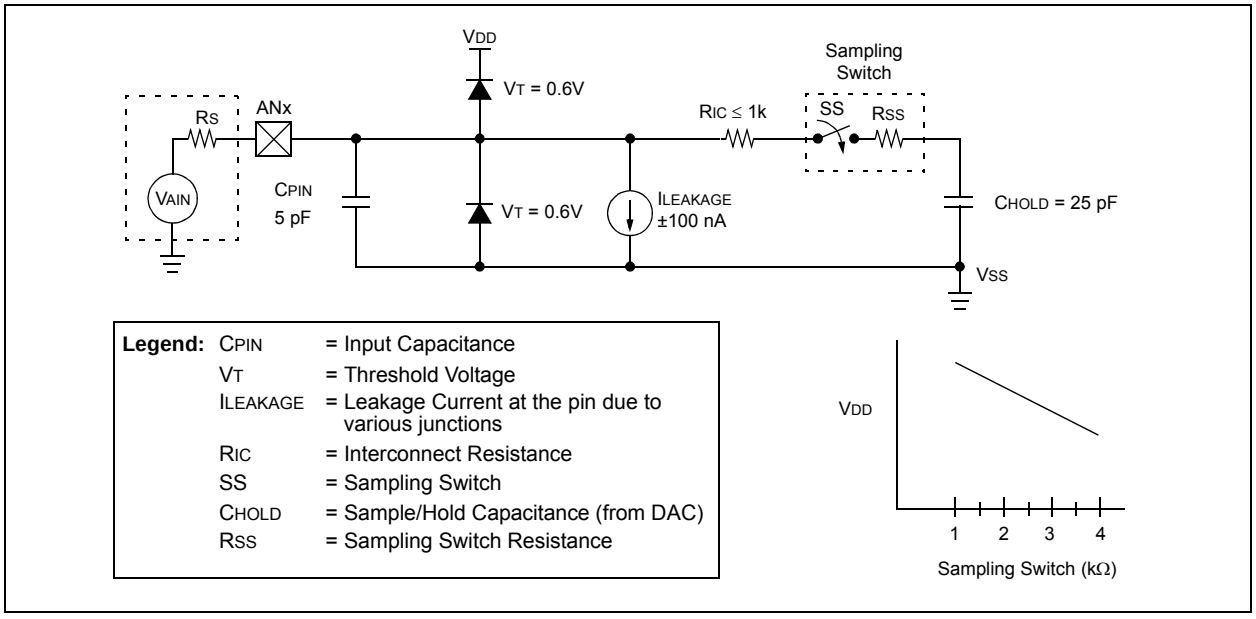
To do an A/D conversion, follow these steps:

1. Configure the A/D module:
   - Configure the required A/D pins as analog pins (ANCON0 and ANCON1)
   - Set the voltage reference (ADCON1)
   - Select the A/D positive and negative input channels (ADCON0 and ADCON1)
   - Select the A/D acquisition time (ADCON2)
   - Select the A/D conversion clock (ADCON2)
   - Turn on the A/D module (ADCON0)
2. Configure the A/D interrupt (if desired):
   - Clear the ADIF bit (PIR1<6>)
   - Set the ADIE bit (PIE1<6>)
   - Set the GIE bit (INTCON<7>)
3. Wait the required acquisition time (if required).
4. Start the conversion:
   - Set the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL) and, if required, clear bit, ADIF.
7. For the next conversion, begin with Step 1 or 2, as required.

   The A/D conversion time per bit is defined as T_{AD}. Before the next acquisition starts, a minimum wait of 2 T_{AD} is required.

FIGURE 23-5: ANALOG INPUT MODEL



REGISTER 27-11: TXBnDLC: TRANSMIT BUFFER 'n' DATA LENGTH CODE REGISTERS [0 ≤ n ≤ 2]

U-0	R/W-x	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0
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 **TXRTR:** Transmit Remote Frame Transmission Request bit
1 = Transmitted message will have the TXRTR bit set
0 = Transmitted message will have the TXRTR bit cleared

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **DLC<3:0>:** Data Length Code bits
1111 = Reserved
1110 = Reserved
1101 = Reserved
1100 = Reserved
1011 = Reserved
1010 = Reserved
1001 = Reserved
1000 = Data length = 8 bytes
0111 = Data length = 7 bytes
0110 = Data length = 6 bytes
0101 = Data length = 5 bytes
0100 = Data length = 4 bytes
0011 = Data length = 3 bytes
0010 = Data length = 2 bytes
0001 = Data length = 1 bytes
0000 = Data length = 0 bytes

REGISTER 27-12: TXERRCNT: TRANSMIT ERROR COUNT REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
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 7-0 **TEC<7:0>:** Transmit Error Counter bits
This register contains a value which is derived from the rate at which errors occur. When the error count overflows, the bus-off state occurs. When the bus has 128 occurrences of 11 consecutive recessive bits, the counter value is cleared.

27.7 Message Reception

27.7.1 RECEIVING A MESSAGE

Of all receive buffers, the MAB is always committed to receiving the next message from the bus. The MCU can access one buffer while the other buffer is available for message reception or holding a previously received message.

Note:	The entire contents of the MAB are moved into the receive buffer once a message is accepted. This means that regardless of the type of identifier (standard or extended) and the number of data bytes received, the entire receive buffer is overwritten with the MAB contents. Therefore, the contents of all registers in the buffer must be assumed to have been modified when any message is received.

When a message is moved into either of the receive buffers, the associated RXFUL bit is set. This bit must be cleared by the MCU when it has completed processing the message in the buffer in order to allow a new message to be received into the buffer. This bit provides a positive lockout to ensure that the firmware has finished with the message before the module attempts to load a new message into the receive buffer. If the receive interrupt is enabled, an interrupt will be generated to indicate that a valid message has been received.

Once a message is loaded into any matching buffer, user firmware may determine exactly what filter caused this reception by checking the filter hit bits in the RXBnCON or BnCON registers. In Mode 0, FILHIT<2:0> of RXBnCON serve as filter hit bits. In Mode 1 and 2, FILHIT<4:0> bits of BnCON serve as filter hit bits. The same registers also indicate whether the current message is an RTR frame or not. A received message is considered a standard identifier message if the EXID/EXIDE bit in the RXBnSIDL or the BnSIDL register is cleared. Conversely, a set EXID bit indicates an extended identifier message. If the received message is a standard identifier message, user firmware needs to read the SIDL and SIDH registers. In the case of an extended identifier message, firmware should read the SIDL, SIDH, EIDL and EIDH registers. If the RXBnDLC or BnDLC register contain non-zero data count, user firmware should also read the corresponding number of data bytes by accessing the RXBnDm or the BnDm registers. When a received message is an RTR, and if the current buffer is not configured for automatic RTR handling, user firmware must take appropriate action and respond manually.

Each receive buffer contains RXM bits to set special Receive modes. In Mode 0, RXM<1:0> bits in RXBnCON define a total of four Receive modes. In Mode 1 and 2, RXM1 bit, in combination with the EXID mask and filter bit, define the same four receive modes. Normally, these bits are set to '00' to enable reception of all valid messages as determined by the appropriate acceptance filters. In this case, the determination of whether or not to receive standard or extended messages is determined by the EXIDE bit in the acceptance filter register. In Mode 0, if the RXM bits are set to '01' or '10', the receiver will accept only messages with standard or extended identifiers, respectively. If an acceptance filter has the EXIDE bit set, such that it does not correspond with the RXM mode, that acceptance filter is rendered useless. In Mode 1 and 2, setting EXID in the SIDL Mask register will ensure that only standard or extended identifiers are received. These two modes of RXM bits can be used in systems where it is known that only standard or extended messages will be on the bus. If the RXM bits are set to '11' (RXM1 = 1 in Mode 1 and 2), the buffer will receive all messages regardless of the values of the acceptance filters. Also, if a message has an error before the end of frame, that portion of the message assembled in the MAB before the error frame will be loaded into the buffer. This mode may serve as a valuable debugging tool for a given CAN network. It should not be used in an actual system environment as the actual system will always have some bus errors and all nodes on the bus are expected to ignore them.

In Mode 1 and 2, when a programmable buffer is configured as a transmit buffer and one or more acceptance filters are associated with it, all incoming messages matching this acceptance filter criteria will be discarded. To avoid this scenario, user firmware must make sure that there are no acceptance filters associated with a buffer configured as a transmit buffer.

27.7.2 RECEIVE PRIORITY

When in Mode 0, RXB0 is the higher priority buffer and has two message acceptance filters associated with it. RXB1 is the lower priority buffer and has four acceptance filters associated with it. The lower number of acceptance filters makes the match on RXB0 more restrictive and implies a higher priority for that buffer. Additionally, the RXB0CON register can be configured such that if RXB0 contains a valid message and another valid message is received, an overflow error will not occur and the new message will be moved into RXB1 regardless of the acceptance criteria of RXB1. There are also two programmable acceptance filter masks available, one for each receive buffer (see **Section 27.5 "CAN Message Buffers"**).

In Mode 1 and 2, there are a total of 16 acceptance filters available and each can be dynamically assigned to any of the receive buffers. A buffer with a lower number has higher priority. Given this, if an incoming message matches with two or more receive buffer acceptance criteria, the buffer with the lower number will be loaded with that message.

MULLW **Multiply Literal with W**

Syntax: MULLW k

Operands: $0 \le k \le 255$

Operation: (W) x k → PRODH:PRODL

Status Affected: None

Encoding:

0000	1101	kkkk	kkkk

Description: An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in the PRODH:PRODL register pair. PRODH contains the high byte.

W is unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL

Example: `MULLW 0C4h`

Before Instruction
W = E2h
PRODH = ?
PRODL = ?
After Instruction
W = E2h
PRODH = ADh
PRODL = 08h

MULWF **Multiply W with f**

Syntax: MULWF f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (W) x (f) → PRODH:PRODL

Status Affected: None

Encoding:

0000	001a	ffff	ffff

Description: An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

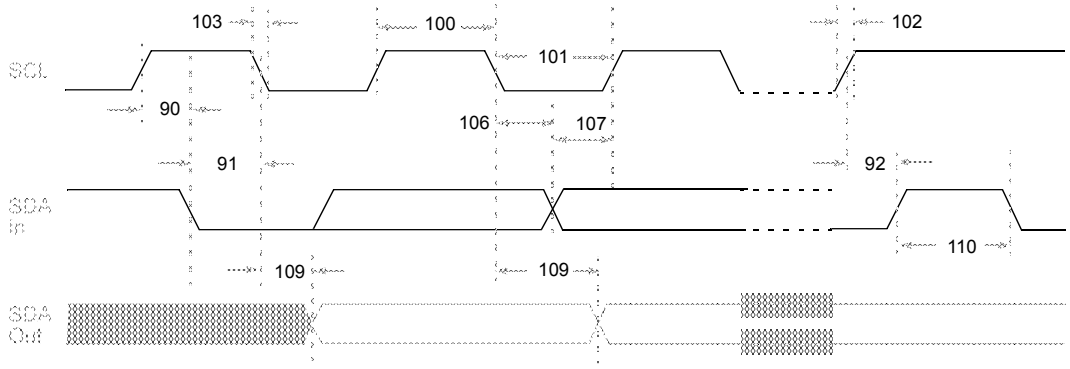
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL

Example: `MULWF REG, 1`

Before Instruction
W = C4h
REG = B5h
PRODH = ?
PRODL = ?
After Instruction
W = C4h
REG = B5h
PRODH = 8Ah
PRODL = 94h

FIGURE 31-16: I²C™ BUS DATA TIMING



Note: Refer to Figure 31-3 for load conditions.

TABLE 31-20: I²C™ BUS DATA REQUIREMENTS (SLAVE MODE)

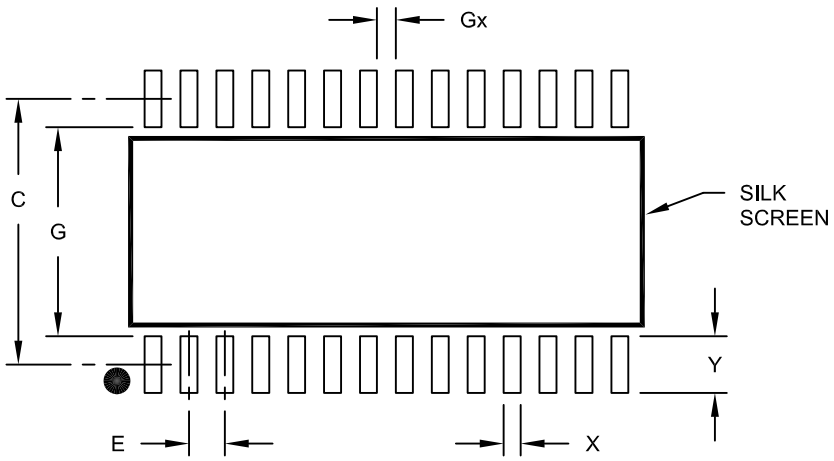
Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	4.0	—	µs	
			400 kHz mode	0.6	—	µs	
			MSSP module	1.5 TCY	—		
101	TLOW	Clock Low Time	100 kHz mode	4.7	—	µs	
			400 kHz mode	1.3	—	µs	
			MSSP module	1.5 TCY	—		
102	TR	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
103	TF	SDA and SCL Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	µs	
91	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	µs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
D102	CB	Bus Capacitive Loading		—	400	pF	

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode I²C™ bus device can be used in a Standard mode I²C bus system, but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

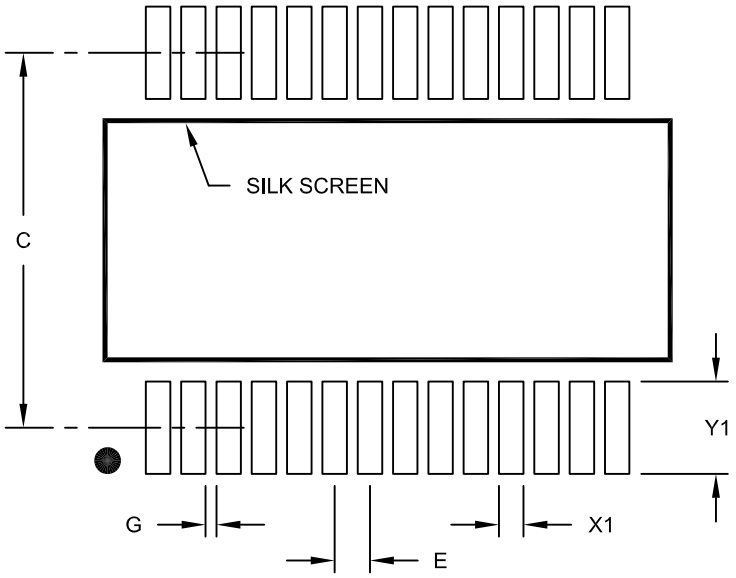
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A